Aalto University
School of Engineering

Lauri Paukkunen

Development of metrics and automation for product model verification

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology.

Espoo, 15th January 2018
Supervisor: Professor Matti Pietola
Instructor: M. Sc. Petri Pohjoispuro

Aalto-yliopisto
Insinööritieteiden
korkeakoulu

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi

Diplomityön tiivistelmä

Tekijä Lauri Paukkunen		
Työn nimi Tuotemallien tarkistuksen metriikan kehitys ja automaatio		
Koulutusohjelma Konetekniikka		
Pää-/sivuaine Koneensuunnittelu/Mekatroniikka		**Koodi** K3001
Työn valvoja Prof. Matti Pietola		
Työn ohjaaja(t) DI Petri Pohjoispuro		
Päivämäärä 15.01.2018	**Sivumäärä** 65+7	**Kieli** Englanti

Tiivistelmä

Tuotteiden laatuun on jo pidemmän aikaa kiinnitetty paljon huomiota insinööriprosesseissa ja tutkimuksessa. Myös tuotemallien laatu voidaan nähdä insinöörityön kannalta elintärkeässä asemassa, erityisesti systeemeissä jotka perustuvat alaspäin virtaavaan tietoon. Mallien laatu vaikuttaa muun muassa sen tarkkuuteen ja muokattavuuteen sekä koko mallinnus- ja suunnittelujärjestelmän ketteryyteen. Huolellinen ja läpikotainen tarkistus on tärkeä osa tuotemallien laadun kehittämistä. Mallien manuaalinen tarkastaminen voi olla työlästä ja aikaavievää. Käyttämällä automaatiota tarkistuksen apuna, voidaan saavuttaa etuja tarkistuksen nopeudessa ja lopputuloksessa.

Tämän diplomityön tavoitteena on kehittää tuotemallien tarkastuksen metriikkaa ja automaatiota. Metriikan kehitys perustuu kirjallisuustutkimukseen sekä muun muassa haastatteluissa kartoitettuihin yrityksen tarpeisiin. Tavoitteena on luoda tuotemalleille metriikkaa, joita vasten niiden ominaisuuksia voidaan arvioida. Myös tarkistuksen automaatiota tutkitaan ja tavoitteena on luoda automaattinen työkalu, jota voidaan käyttää yrityksen tämän hetkisessä suunnittelujärjestelmässä.

Tutkimuksen lopputuloksena syntyi lista tuotemallien laadun ulottuvuuksista niihin liitetyillä metriikoilla ja metriikan mukainen PTC ModelCHECK tarkistuspohja 3D-malleille, joka löytyy automaattisesti virheitä malleista. ModelCHECK valittiin työkaluksi, koska se on valmiiksi saatavilla yrityksen nykyisessä mallinnusjärjestelmässä, joilloin automatisointi on erittäin kustannustehokasta.

Avainsanat Tuotemalli, Tarkistus, Verifiointi, Mallinnus, Laatu

Author Lauri Paukkunen

Title of thesis Development of metrics and automation for product model verification

Degree programme Machine Design

Major/minor Machine Design/Mechatronics **Code** K3001

Thesis supervisor Prof. Matti Pietola

Thesis advisor(s) M.Sc. Petri Pohjoispuro

Date 15.01.2018 **Number of pages** 65+7 **Language** English

Abstract

A lot of interest and research has been focused on product quality and it is recognized as a crucial aspect of engineering. The quality of product models can also be seen as essential in engineering workflow especially in systems based on downstream data. Model quality effects not only the models accuracy and modifiability but also the agility of the whole engineering systems. Careful and thorough verification plays an important part in effecting product model quality. Verifying product models and designs manually can be laborious and time-consuming process. By automating parts of the verification process, benefits can be seen in the time frame and end results of the verification.

The goal of the thesis is to develop metrics and automation for product model verification. Development of metrics is executed by researching literature for model quality metrics and construct a set of metrics for the company. Furthermore, the possibilities of product model verification automation are studied and a working automated model verification tool shall be created based on the metrics. The tool is intended be used in the current modeling environment.

The outcomes of this thesis are a list of product quality dimensions with their corresponding metrics and a customized PTC ModelCHECK check that can automatically identify issues in product models. Quality dimensions were identified based on company needs and literature research. ModelCHECK platform was chosen for verification tool development as the software is readily available for the company which means it is a cost-effective way of utilizing automated product model verification in current design environment.

Keywords Product model, Verification, Quality metrics, Automated verification

Preface

I would like to thank KONE for the opportunity to write this master's thesis and supporting me in the beginning on my professional career. I would especially like to thank my thesis supervisor Professor Matti Pietola and my instructor M.Sc Petri Pohjoispuro. A huge appreciation also goes to my friends, family, co-workers and fellow students for supporting me on my educational and professional paths.

Espoo 15.01.2018

Lauri Paukkunen

Lauri Paukkunen

Contents

Abbreviations

3DCAD	Three-dimensional Computer Aided Design
A-process	Standard engineering process inside the product platform
API	Application Programming Interface
BIM	Building Information Model
BOM	Bill of materials
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacturing
C-process	Customer specific engineering process outside of product platform
CWS	Configuration Validation and Verification Suite
ERP	Enterprise Resource Planning
MBD	Model-Based Definition
MQT	Model Quality Tools
PDM	Product Data Management
PLM	Product Life-Cycle Management
R&D	Research and Development
TQM	Total Quality Management
V&V	Verification and validation
Verification	*"Creating the model right"*
Validation	*"Creating the right model"*
XML	eXtensive Markup Language

1 Introduction

1.1 Research background

A lot of interest and research has been focused on product quality and it is recognized as a crucial aspect of engineering. The quality of product models can also be seen as essential in modern product life-cycle management (PLM) systems and in engineering overall. This means that careful and thorough verification also plays an important part in addition to defining modeling methodology. One additional element that buffs the importance of model quality is configurability. KONE product models are highly configurable, and thus accurate and robust initial models/data are essential to minimize additional work.

In this thesis, product model will be defined as combined PDM-model (Product data management) and 3DCAD model (Computer Aided Design). PDM models can also be considered as data models and 3DCAD models as geometric representation with metadata. Data model verification as a field has a lot of studies and these finding can be fitted to PDM-model verification. Some studies of 3DCAD-model verification also exist but they can also be though as pure mathematical models for which more studies can be found. Also, studies of software verification were researched since they work on similar ideas.

Automatic verification tools can be found in many design and CAD software already but they are usually have limited potential, are not fully utilized or are even ignored altogether. Benefits of automation are usually quite clear: it aims to eliminate repetitive and time consuming tasks from employee's workload, creating more time for more productive tasks. As the product design in mechanical engineering is also going towards automated design and model creation, other design task will also become points of interest.

Benefits of product quality are also quite well documented. Better product data quality will help all around the process as the data is used as a starting point of multiple different actions and tasks. Accurate and robust models enable high-level automation throughout the whole design and development process. PLM systems require common data between different systems to work correctly.

1.2 Research questions

This thesis answers the question:

"What metrics can be used to evaluate and verify product models?"

This should include comments on what metrics should be evaluated in a model verification process in the company. As an additional question, the automation possibilities of the verification are researched. This question can be listed as:

"Can product model verification be automated?"

If model verification can be automated, thesis should also give answers on how to do it, on what level can it be done and on what level should the automation be taken.

1.3 Goals of the thesis

This thesis has two primary goals. First goal of the thesis is to identify and select suitable metrics for verification of product models at KONE. This means selecting metrics that can be used effectively in product verification and to use them also for supporting future needs.

The second goal is to develop a process for automated verification. Focus on automation is with using tools currently available in the modeling environment. In 3DCAD, internal tools such as ModelCHECK embedded in PTC's Creo 2.0 design software shall be investigated. Furthermore, in PDM the current functionalities are explored. Goal of the automation research is to create a working tool for verification. Successful tool can be used as a starting point for future development in the automating product model verification and implementing the verification tools in the production environment. One additional goal is to review the current product model verification and the new guidelines with use cases.

1.4 Scope of the thesis

The scope of the thesis is verification of product models in 3DCAD and PDM environments. For example, verification of Building Information Models (BIM) is left out, although researching this would also be beneficial for the company. Also on CAD models, the scope is limited to 3D side since the company and engineering work in general is moving away from pure 2D design. Similar methods could possibly be implemented for verifying work drawings and other 2D documents but theoretically these are derived from 3D models thus making at least the original data correct if the 3D model is verified. In PDM, the scope is to identify metrics for PDM model verification.

On automation side, this thesis focuses on automating and configuring currently available tools and processes instead of creating new tools to minimize the workload required for the change. No changes will be made on the current PDM system, but automation of PDM model verification shall be investigated using current external software/communication interfaces. The thesis is focused on verification of models that can be described as *"creating the model right"*. This means that models should be created according to guidelines and should contain the given data accurately. This thesis will not discuss model validation, described as *"creating the right model",* in other words comparing the model to a real-life counterpart.

1.5 Methods

For the evaluation and development of verification metrics, literature research will be conducted. The study of verification automation includes both literature research and creating a functional verification platform or process that can be implemented in current design environment and for future work. Case examples shall be used for comparison between current process and the suggested changes. Interviews and benchmarking of verification procedures in other industries shall be used to gain more information about verification practices inside and outside of the company.

2 Product modelling

2.1 3DCAD

3D-modelling has become the driving development factor in mechanical engineering in recent years. Even though still surprisingly large part of engineering work is done in 2D, the 3DCAD software are catching quickly. As the computing power of processors have risen, the 3DCAD software have been able to provide more and more and the differences to 2D software are becoming bigger and bigger. 3D modelling can also produce 2D drawings almost automatically so, nothing is lost in that sense.

2.1.1 Benefits of 3D-modeling

One of the biggest benefits of 3D-modeling in comparison to 2D-modeling is the increased visuality. When the products are modeled directly in 3D, the product should look exactly as the model does on the designer's computer. This helps with tracking down errors in the design in the earliest possible phase. [1]

Another distinguished advantage of using 3D-modeling is the different ways of modelling inside the software. Most of the modern 3D-modeling software use boundary representation (b-rep) modelling, sheet metal modelling and surface modelling. Major part of models in KONE are b-rep volume models and sheet metal models. Sheet parts are used especially with certain components such as the elevator car, which have a lot of metal paneling. 3D models can also be given mechanical and physical properties such as mass and material parameters. These can be used to automatically calculate for example a mass of an assembly or used for other calculations. [1]

Parametric and feature based modeling further benefit the use of 3D-models. They enable flexible models that can be re-used or modified without much of a rework. Also, associativity is a common feature in modern CAD programs. This means that the features can have references and relationship to each other's which is essential with configurable product models. With parametric 3DCAD software, the model parameters such as dimension can be easily changes resulting in agile and changeable models. This can also further be used for automated creation of models with using script, internal software tools or APIs. [1][2]

3D-models also make creating assemblies a lot easier. Full assemblies can be constructed from hundreds or even thousands of parts and subassemblies to represent the real-life product as is. Some modelling programs also enable using so called simplified representations that only show certain components of geometries in the assembly. This can drastically decrease the generation times of big models and ease the work layout and other work phases where full detailed models are not necessarily needed. Using the full representation alongside with simplified representations can streamline the modelling work significantly. [1] [2]

In modern 3D-modelling, skeleton models are also often used. They are a useful and productive way of taking advantages of working in the 3D-environment especially with configurable and complex products. By using model skeletons, it is possible to create even more simplified versions of assemblies and to point the references of different part to these skeleton structures. This enables highly configurable and modular product designs to be effective

and easier to comprehend. Furthermore, centralizing the references to skeleton models instead of scattered references throughout the model help with reference control and management. [1] [2]

2.1.2 CAD representations

There are two primary representation schemas in mechanical CAD applications: Boundary Representation (B-Rep), and history-based parametric feature-based models. However, in actual design work and the real-life use, these different schemas are usually interconnected, for example in feature-based CAD models. Solid models can be defined by the enclosed shells for the material, closed meaning the surface of the model is divided to distinct internal and external volumes. In contrast, closed B-rep and meshed models provide a complete representation of a solid shape but do provide information on how the shape was created. [2]

Representations are explicit when their details are immediately available without the need for any calculations. Representations are procedural or history-based, if they are described in terms of a sequence of procedures. ISO 10303-42 [3] defines explicit product shape models as a fully detailed models of the boundary representationor related type. B-rep is a particular type of explicit representation where suitable sets of connected geometric elements are used to represent the vertices, edges and faces for the boundaries for the solid model. The exact shape and positions in space are defined by geometric information and the links between elements is defined by topological information. There are also further classifications for B-Rep representations such as faceted B-rep and boundary curve based. Furthermore, hybrid representations are combitions of both explicit and procedural representation methods. [2][3]

Procedural modeling techniques such as history-based parametric modeling create the 3D models for the sets of rules established. Procedural models aim to capture all or at least part of the design intent. This is beneficial for relaying design intent and makes them easier to modify in the future. In procedural modeling, the approach is to CAD models to include feature-based design and constraint-based modeling. Feature-based models are especially useful when using feature-based downstream applications such as numerical control machining. [2][4]

2.1.3 3DCAD environment

Current modeling environment at KONE consist of Creo 2.0 CAD software and Windchill 10.2. Windchill is used only for storing and managing 3D models. PTC Creo is a fully functional feature based parametric 3DCAD software. The parametric nature is based on using constraints, references, changeable dimension and rule set for features and dimensions. Creo can be used for solid, surface and sheet modeling. The program itself offers some low-level model configuration tools such as family tables and the use of parameters and ProProgram for creating configurable model structures. However, at KONE a customized interface inside of Creo is primarily used to help create configurability in unison to company's PDM system. In engineering use, the 3DCAD software are rarely independents parts of the engineering system. To facilitate, control and store data different type of product management systems are used.

In 3D modeling, components are designed using multiple assemblies, parts and standard parts. Individual elevator or escalator product lines form their own platforms that includes all the options used in the corresponding product line. Using off-platform components or

special design require so called C-process engineering which is explained more in-depth in paragraph 2.5.3. Some modeling work such as creating some of the elevator layout drawings is still done in the 2D environment. This can be helpful in manipulating blocks in 2D drawings but is not the preferred way of working as its can lead to other problems. As disclosed earlier, 2D is not in the scope of this thesis and thus will be mainly ignored.

2.2 Product data management

2.2.1 Product data

Product data refers to all data the product is involved with from design to manufacturing. It varies largely between different companies and needs. Ideally all information related to the product would be under the corresponding product data model. In reality, data is often scattered across and inside the system, missing altogether or existing only on a certain designer's head or a piece of paper nowhere to be found. [5]

Some of the main sectors of product data are the detailed product specification, test and analysis data, manufacturing data and documentation. The format of the product data itself can vary from parameters in the system to technical documentation, CAD and simulation software files and product configurability rules to name a few. Detailed product specification should include the product structure (including Bill-of-Materials, BOMs) and the parameter and attribute data as well as other metadata. Technically attributes don't differ from product parameter data in any way but are usually separated due that they are often used differently in the system. Attributes can be defined as static information instead of changeable parameter values. Manufacturing data can for example consist of vendor information, product delivery structures, manufacturing information and drawings and other related data. Test data includes data for example from CAE tools such as FEM calculations. Other data aspects included in PDM-models are configuration rules and scripts, delivery structure (delivery BOM), documentation, test/calculation data and more. [6]

PDMs can also store data for individual configurations, which are different variants of the master engineering structure of the product. PDM is also used for revisioning and change management. For revisioning, minor and major revisions exist in the software. PDM systems also track the release state of items. With fully configurable products, PDM-models are much more complex. They need to include all the parameter needed for the product and rules for the configuration and part selection. With simple components, these rules can be simple but in a case of complex product assemblies with high configurability the rules can include input and output data from multiple sources and calculations for others parameters. This complex system of rules makes it very hard for designers to verify the rules and that they are working. [7][8]

Another metadata aspect of a product model is order data. Order data concludes for the order parameters which are primarily provided from customers and sales organization via sales tools such as sales configurators. While the lifecycle of order goes further more data is added to the PDM order structure. The engineering BOM can be created automatically from the order parameters using a PDM configurator. Changes can be also made manually to the engineering structure and thus modified also in c-process cases. The delivery structure DS can be created based of the engineering BOM. Order data can also include documents and other data. [6]

2.2.2 PDM systems

PDM is a system that manages the company's processes and the data related to them. One of the most important tasks of PDM is to increase the internal communication and to promote common data to be used throughout the company. PDM systems unifies the term, processes and the identification data for the product and so defines the common language about the product inside the company. Product data management can also be divided to different sub categories such as: name management, document management, product structure management and change management. PDM system consist of multiple different components and functionalities such as information warehouse. Some notable functionalities for PDM are working as information warehouse and managing it, defining the product and workflow structures, workflow controlling system and to provide and the infrastructure for all this. [6][7]

PDM systems are essential in modern engineering environments and are often directly implemented in the design software. They are mostly used in companies to store and share data between different segments of the company, subcontractors and sometimes even with customers. They hold the essential data for the products, sometimes throughout the product life-cycle. It can also be used to store models and other files relating to the product such as ordering and engineering instructions. PDM systems are often multifunctional platform that are used from sales to R&D. The information flow in PDM systems can be either one way or two-way. The advantages in two-way systems is that changes can be made also in PDM. Product data management systems helps companies to keep the product data correct and up-to-data. Especially helpful in this aspect is the capability of revisioning product models. Using different classes can be used to create and manage product families and many systems can even configure new product variants based on user given data. New variants can also be specifically created for orders to manage order bound product models. [7]

Currently KONE is using Variantum's VariPDM product for PDM functions in the modeling environment. Since KONE is one of their biggest clients, modification and customization of the user interface and functionalities is possible and the PDM system is still evolving. The PDM system itself works as a master data based in the product work flow. Its purpose is mainly to provide better configurable product structures. Packaged with Creo 2.0, the company also has Windchill 10.2 available. However, the system in not used for actual PDM functionalities but it is the main platform for storing the product 3D-model data. This means that the product model data is currently scattered between two or three platforms which should be considered when developing the system in the future.

2.2.3 PLM systems

Product Life-cycle Management systems aim to handle the full life-cycle of the product starting from R&D to maintenance and scrapping, thus making them more of a uniform platform for companies. As Stark [6] suggests, PLM enables information automation and system integration with accurate and timely product data. Furthermore, we can deduct that PLM system itself isn't a magic wand for product management and quality concerns but when used with proper data can lead to automated information load. Many modern 3DCAD software implement PLM solutions but in large companies, these solutions are usually modified according to the customer's needs. It is also possible to use different software vendors for different fields in product management by utilizing for example APIs and neutral file types. [6]

PLM can be used to eliminate so called separate islands of automation. Island of automation describe different applications in the process that must be able to work independently from other applications. Wasteful duplication of functionality can be found in many engineering systems. Also, minimizing the interfaces between systems is helpful in many ways. This may lead to decrease in errors across segments, shorten design times and increased usability. PLM enables information automation and system integration with accurate and timely product data. PLM and PDM systems share a lot of similarities but there are also some fundamental differences. PDM systems are more focused on the engineering process, whereas PLM systems are more capable of cross-business processes. The same applies with product information: PDM has engineering viewpoints of product data whereas PLM attempt to contain the engineering, supply chain and commercial views of the product. [6]

2.3 Communication between systems

2.3.1 Need for common data

Without proper sharing of the data, multiprogram systems are basically useless. Information flow between different departments and units has been seen to have positive effect in product quality and lessen duplicate work. Communicating data forward becomes even more important in the future as system evolve into doing more complex tasks. The need for common data is undisputable and a desired goal in PLM systems. Differences between parameters or attributes in different systems can cause design, production and even product safety issues. While handling and uniforming vast amounts of data is difficult, efforts should be made to get the data as perfect as possible. Many companies have legacy data, which basically describes product in an older format and possibly lacks some information. In PLM systems, the conversion of the legacy data is also very important. [6][7][9]

Common data bring benefits on multiple levels for designers, managers, subcontractors, maintenance and frontline operations. Also from the system point of view the common data brings benefits such as revision and characteristics control. With common data, future changes and updates in the engineering systems become easier to implement, as the data should be the same in different systems. [6]

2.3.2 Methods of communication

Communication between systems inside the engineering work is very important aspect in a good engineering design environment as was discussed in the earlier chapter. As the PLM functionalities in some engineering environments are divided between tools, the communication between systems and sharing product data is crucial for the process.

Most modern-day modeling software include APIs (Application Programming Interface). These can be used for inter-tool communication as well as creating new tools inside the modeling software. Some software APIs are available for all users, but using them requires expert knowledge of both systems. In best case scenario, the desired API applications are embedded in the software used and feel like a normal part of the software for the users. In KONE, a third-party application embedded to Creo is used for communicating between PDM and 3DCAD systems.

Another commonly used option is to relay product data is with files such as XLS or XML. XMLs are one of the most popular choices for relaying information between different engineering systems. They are also widely used in KONE and for example are the primarily

method of exchanging information about parameters and position number in assemblies between CAD and PDM software. The files themselves are often custom formalized so they are almost always company specific. There are some industry specific XML standards such as RailXML but for example the elevator industry does currently not have one. Without encryption, the data is still accessible and can be converted to different formalized versions which means that backwards compatibility exist also with design environment changes to some extent. Even TXT files can be used to exchange information and it is up to the companies to choose a method and format suitable for them and their environment. [8][10]

Attempts have been made for a neutral international standard for relaying model data. These formats include STEP, IGES, SDAI, PDF and STL but even if many of these formats are used in relaying data between different systems, at the moment, there is no all-encompassing neutral standard for relaying all the product model data.

2.4 Product configurability

2.4.1 Configurability of product models

Configurability is based on product parameters, often called product characteristics. These can have either numerical, Boolean or string values depending on the type. In order to create configurations, a so-called superBOM structure is needed to disclose the parameter ranges and different possibilities. SuperBOM term is used to describe the complete product structure, including different options in case of modular or configurable product. It should contain all the elements described in the product platform. A set of characteristics is used to create a new *configuration* or *variant* of the main product structure. [1]

Configuration knowledge represents the rules of generating product variants and constraints that variants should satisfy [11]. For example, if you build a car configuration system, you must define generally what a car is in that context. A car has an engine, four wheels and a spare, trim and so on. There are also constraints between different part of the car, for example the horsepower of the engine and the size of the wheels. All the products characteristics are not necessarily included as configuration characteristics. Additional characteristic can calculated by defining rules, references and parameter tables for the product models. These are usually defined in 3DCAD and PDM environments. [11]

Product configurators can be found in PDM, PLM and CAD system and as standalone software. Figure 1 presents an example of a user interface for a product configurator. The example show a very user-friendly product configurator, whereas usually configurators are more bareboned and embedded to the existing systems such as 3DCAD or PDM. Product configurator are not only used in engineering works but can be implemented as a direct ordering interface between the customer and the company as for example some kitchen manufacturers do. The configurator usually provides some sort of visualization of the product to help assess the properties of the product. [8]

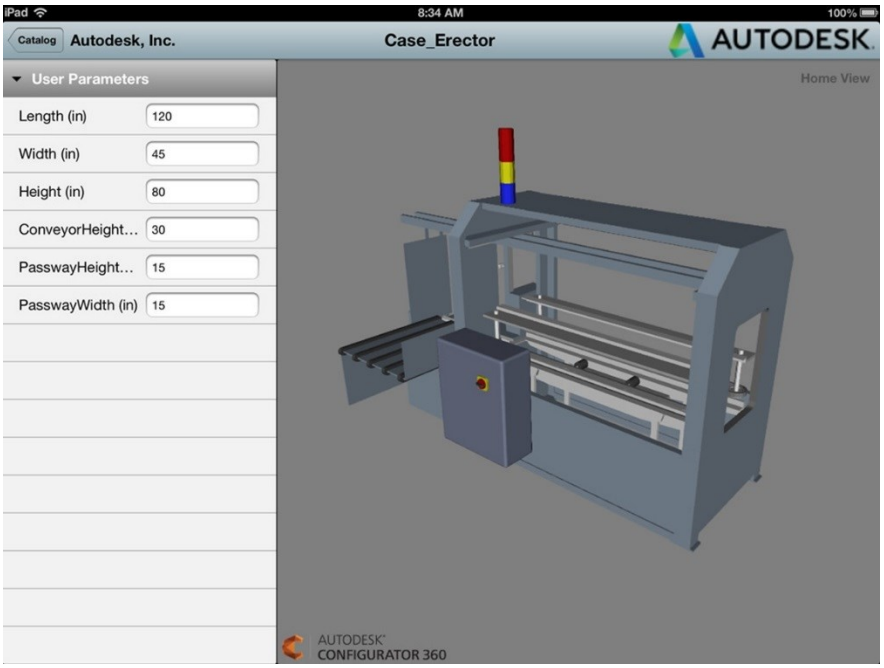


Figure 1. Example of user interface for a product configurator [11].

KONE product models are overall highly configurable. Buildings are very different and there are huge differences in customer needs. For example, the car load capability affects multiple other components. This also brings additional requirements for the product models to be used effectively and reliably.

2.4.2 Massconfigurability and Engineering-to-order

Massconfigurability describes the ability of the model to be easily configured in to different variants for example by using model automation. For product to be mass configurable it must be able to take characteristic values and configure accordingly. Modularity is a good feature but not necessary for mass configurable model. For massconfigurability cases, it is not enough that modifications are easy to make but making them should be possible in mass scale and preferably fast. An industry example of massconfigurability tool is a product configurator. In case of massconfigurability, re-usability of the models is extremely important. With massconfigurable models, the same model can be configured with different parameter without making changes to the high-level product model. This means the model can be re-used theoretically infinite times. The actual re-using of the top-level massconfigurable product models themselves is not that important since the product models should be tailor made for different platforms. [13]

KONE product models are created in such a way that creating different configurations of product is easy. In a case where only the starting characteristics and components within the same product platform are used or changed this is considered to be "A-process" product. This means that the configuration, 3D-model and delivery structure can be created automatically based on the order parameters. In the current modeling environment, configurable products sometimes must be configured multiple times to get the result wanted, which is of course not desirable. This problem is might be a result of using non-standard modelling and referencing, resulting in structure that is not a genuine top-down structure. In theory, an authentic top-down structure should configure correctly with one run, since all the infor-

mation flows downwards. In problem cases, references are possibly created in an incompatible manner in specific product level or between multiple product structure levels. [mass] [14]

In some cases, the top-level product models or platforms do not offer enough flexibility or a special component needs to be designed for specific orders. This process is often referred to as engineer-to-order. At KONE, this type of customer specific configuration of the product model is considered to be "C-process" design. The workflow differs significantly from A-process products as customer specific products are usually "hand-made". This means that original model can be changed in a way that defies the platform or product rules and/or is not possible to create by normal configurations. This requires design engineer to manually alter the product characteristics and structures. The customer specific changes to the main product models can be costly and time consuming. Furthermore, the product models from the c-process are usually not easily re-usable as such. C-process designs should preferably be considered already while creating the massconfigurable A-process design. Creating models in a flexible way early on would ease the modification and customization workload with both A- and C-process. [15]

2.5 Future trends in engineering processes and modeling

Hirz et al. [16] present that there is a distinct trend to increasingly change from purely geometry creation to integrated development cycles including layout and simulation. This means utilizing also the simulation and calculation tools directly in the modeling software where they are easily available for the designer. 3D-modeling software can conduct for example mass calculations and interference checks, mechanism simulations and even FEM calculations. Even though the internal CAD tools cannot replace the full functional dedicated FEM software, the engineers job can be made easier to make initial calculation and early design optimization. For FEM applications, the quality and accuracy of the mesh structure has a great impact on the results of the analysis. If this model data is downstreamed from CAD environment, the quality of the CAD model should be subject to higher quality standards to ensure that information flows correctly. [16]

In addition to FEM, using multibody simulation (MBS) is also a trend that is continuing to grow. MBS can be used to simulate movement and mechanism of the products in the modeling phase. They are already widely used in automotive engineering and will continue to be implemented more in other industries as well. MBS models can be designed in the simulation software but the most common method is to use 3DCAD model as master model. This requires conversion of formats and STEP-files are usually used for this purpose. Having MBS or FEM tools in the native modeling environment benefit the process significantly as it eliminates the need for model conversion which may lead to defects in the model. [16]

The full capabilities of programming inside CAD environment can be utilized to create smart configurable models and to implement rules for modeling. This means that in the future, automated design features will be a pivotal point in the workflow and is required to function correctly. Suitable high quality 3D-models can be used for creating additional formats and offerings in downstream applications. For example, 3D-models can be also used to create Building Information Models (BIM) and for 2D manufacturing and layout drawings. 3D-models can also be used for rendering product images before the actual manufacturing has even started which provides benefits especially for the sales and marketing segments. In modern 3D CAD software, generic object oriented programming is rising. Object oriented

with soft coded algorithms may be seen as the enabler of new type of completely explicit product model providing high flexibility and a high degree of formalization by leveraging knowledge into the product models. [16]

In PDM, enhancement of cooperative work and integration of computer aided design tools are some of the current trends. Furthermore, conveying the design knowledge and creating product templates can benefit the information flow and design re-using. Re-using product model saves a lot of time by removing duplicated work. One current trend is to pursue integrating and unifying engineering tools as PLM systems [16]. At KONE, some changes and customizations are already being implemented to the modeling software themselves. This approach of customizing the modeling tools is used to bring the designers and modelers as much functionalities in a single program as they need. The focus is to lessen the burden of switching between programs and different datasets and thus hopefully creating better models overall. To receive the full benefits of PLM systems the quality of the models in the system needs to be on an adequate level. [6][16]

Model Based Design (MBD) is a method for product data management, in which all the product information is included in the 3D CAD model. This includes product geometrical models, metadata, 2D-drawings, tolerance data and more. The main purpose of MBD is to provide more efficient and streamlined way of utilizing and creation of product data and to utilize the full advantages of 3D models in comparison to 2D drawings used in traditional engineering design work. MBD is widely regarded as the future of product modeling. MBD improves the workflow by gathering all the product data in a one place, minimizes data migration, opens possibilities to utilize 3D product models and lessens costs. One significant aspect is the possibility of having better quality, up-to-date data available in single form. In MBD, the designer can focus of creating the 3D model and centralize all the product data in the model. For example, 2D layout drawings can then be later created based on the 3D model automatically. Visualization is also a huge benefit in MBD.

Digital MockUp (DMU) is a concept that uses a description of the product, usually a 3D-model for the products full life-cycle. Digital Mockup can be used to describe the complete product and even the functionalities of the product. For creating and managing a complete and functional DMU, engineering views from different section of the products life-cycle are needed. DMUs aim to gather information about the product in the same place and to use them as a virtual prototypes. Virtual prototypes can be used to test and optimize designs without the need for physical model. Whereas DMUs are more of a development help tool and a partial representation of the real product Digital Twins aim to represent the entire real life system. They integrate artificial intelligence, machine learning and software analytics with data to create complete digital simulation models that update and change as do their real-life counterparts. They can be used for monitoring, diagnostics and prognostics to optimize product performance and utilization. Digital twins require accurate product models and product data to work as planned. This might be an issue in converting of product data into digital twin compliant. To fully leverage the possibilities of digital twins, the digital version and the product models must be specifically created in unison with the real product.

Knowledge-based Engineering represents a merging of object programming, artificial intelligence and CAD. KBE can be described as engineering based on knowledge models. A knowledge model uses knowledge representation for the artifacts of the design process instead of programming and database techniques. The ultimate goal of a KBE system is be

able to identify the best design practices and contain the engineering expertise in a knowledge base. They aim to use product and process information for modeling of engineering processes and to utilize the model for automation of the the process. One way of implementing KBE is to layer knowledge-based technology on top of existing CAD and other engineering tools.

3 Model quality and verification

3.1 Model and product data quality

In general, model quality can be seen as a vague and subjective term. Different companies and institutes have different ways of implementing and measuring model quality. Furthermore, different data sets require different set of metrics and quality dimensions also based on their use. Overall, quality of the product is a dimension that needs to be worked from manufacturing to design to management. Illustrated in figure 2, the task of working with data quality is a virtuous cycle. The first step is to define scope, quality dimensions, defect thresholds and goals for wanted quality. Secondly, the quality needs to be measured in comparison to the definitions. The results from the quality measurements can then be analyzed to determine what is the current state of the data. Lastly, the quality needs to be improved as well. Improving requires definition of new dimensions and goals and the cycle starts again. [9][17]

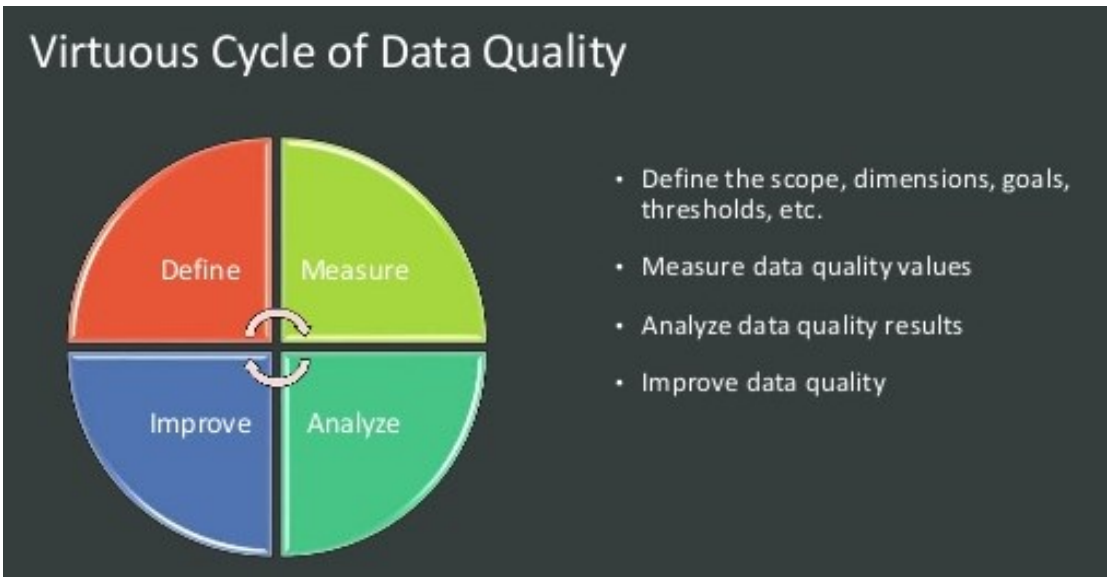


Figure 2. The virtuous cycle of Data Quality [17].

As said, there are multiple ways to describe quality in models. Wand et al suggest that the notion of the data quality depends on the use of data itself. As well as other authors, they define data quality as "fitness for use", showing that the concept of data quality is relative. Furthermore, their research also notes that as important as defining the concept of data quality is to know how it is to be measured. SASIG [19] defines product data quality as a measurement of the accuracy and appropriateness of product data combined with the timeliness with which those data are provided to all the people who need them. This can also be interpreted as that good product quality means providing the right data to the right people at the right time. [18][19]

For some model quality aspects, such as the accuracy of geometrical properties, the quality can be described with standards as is commonly done will physical products. For example, German based automotive industries have created a standard ISO/PAS 26183:2006 [19] which sets the minimum requirements for 3D model accuracy. Furthermore, model mesh accuracy is a quantifiable variable which makes it possible to measure and set thresholds for sufficient quality products. In other situations, where sharing product data is important, correct geometry alone is not enough. Additional organizational information in the product

models must be understood such as naming conventions, layer structure, parameters and more attributes related to the syntactical quality of the model. These aspects become more important as more downstream application are used in the modeling process. [19]

Another important factor in model quality is that the model is created according to the modeling methodology defined for it. This factor might be difficult to control in real industry scenarios, especially in companies that have a wide variety in their products. Bigger variance requires either looser guidelines on the top modelling methodology or additional modelling instruction for different product categories. [20]

3.2 Importance of quality

The scientific and professional communities are unanimous about the importance and the benefits of the quality of physical products. However, the importance of quality CAD and product model data could be argued to be undervalued and often overlooked. Finn [21] proposes that the demands for CAD data should not have different standards as opposed to physical products. Stark [6] suggest that the management needs to make sure that the product data in use is of high quality. Furthermore, without reliable, timely and accurate data, managers and users can't work efficiently. It is also managements duty to enforce product data reuse and to allow it to evolve. Creating high-quality product data is time and resource consuming. Using existing product data however costs much less and helps to reduce time. [6]

The effect of product model and data quality can be seen especially in downstream applications. This means any applications that uses the master model data as a source for product information. Naturally, if the master data is not good quality the working data and the outcome of the downstream application will be the same quality at best without additional data correction steps. This also means that quality is crucial aspect in advanced engineering systems such as PLM, MBE and DMUs. Quality data enables better interfacing between systems in an engineering environment. Using uniform data ease collaboration between design platforms and reduces time and resources spent for converting data. High quality data is also a crucial requirement for example in model automation purposes. Without high quality data to begin with, the whole automation process might become costlier than creating models manually. Furthermore, without sufficient level starting models, the automation only creates bad copies and variants and thus might create more problems than benefits. [6][20]

The quality and the quality control also effect the design times by minimizing the errors and problems that need to solved in the early phases of the product design. As can be seen in figure 3, it is a lot cheaper to make changes in the early points of design. In some cases, especially with big and complex product model, making changes might not even be possible later in the design stages without major rework. Catching errors and quality defects as early as possible in the process create significant benefits with time consumption and costs. [22]

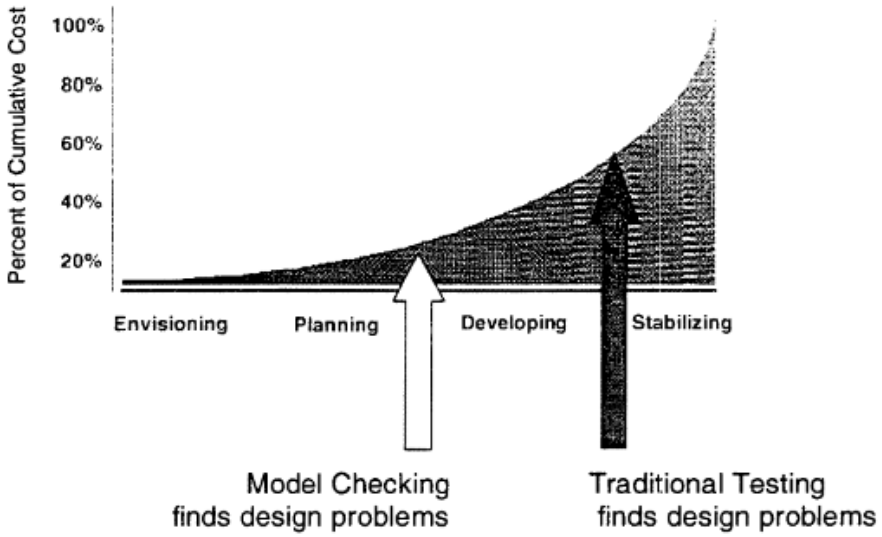


Figure 3. The cost effects of repairing design defects in different phases during the design [22].

Quality also has substantial financial effects. For example, The Data Warehousing Institute [23] reports that poor data quality is costing companies more than $600 billion a year. Furthermore, the cost of repairing or making changes to data or models increases rapidly during the design phase. After the product design phase, making changes will come even more expensive. Mirroring this, it should be a focus from the beginning of the design phase to create high quality and if possible easily changeable models and datasets. [23]

Redman [24] estimates the costs of poor data quality to be as high as 600 Billion in the US alone. The poor quality can be estimated to costs the US economy around 3 Trillion dollars sin a year. Poor data or lack of visibility is cited as a major culprit for project cost overruns. The benefits of improving the quality and usability of data are also significant. It is estimated that if a median Fortune 1000 listed company improved the data usability by 10% the potential increased revenues would be 2 Billion dollars. 46% of survey respondents cite data quality as a barrier for adopting BI/analytics products. Another study estimates that poor data adds 10% to the cost and up to 25% to delivery time in the tooling industry in US. A study examining the economic impact of data exchange problems in German automotive industry calculated as approximately half a billion dollars per year. [24][25][26]

Other often overlooked benefits are easier housekeeping and creating more overall trust for the data in the system. When the data is uniformed it is easier to implement future changes and introduce new tools. With better data available, designers and other users can trust that the data is correct resulting in skipping extra precaution steps or verifying data with experts. These types of benefits usually come along long period of time and thus are usually not considered as important as directly contributing factors in the engineering workflow. [6] [20]

3.3 Product model quality dimensions

Moody et al. [27] proposes that semiotic data quality framework is based on four levels syntactic, semantic, pragmatic and social level. Syntactic level reflects the structure of the data, semantic presents the meaning of the data, pragmatic describes the usage of data and social level concerns the shared understanding of the meaning of symbols. Contero et al. [28] present three levels of quality for CAD models: morphologic, syntactic and semantic. Morphologic quality relates to the geometrical and topological properties, syntactic

measures the proper use of modeling conventions and semantic focuses on models ability for modification and reuse. Product data quality can be also divided to explicit model properties and procedural model properties and further to morphological, syntactic and semantic properties as can be seen in Figure 4. Quality issues can also be classified based on the type of representation. [27][28]

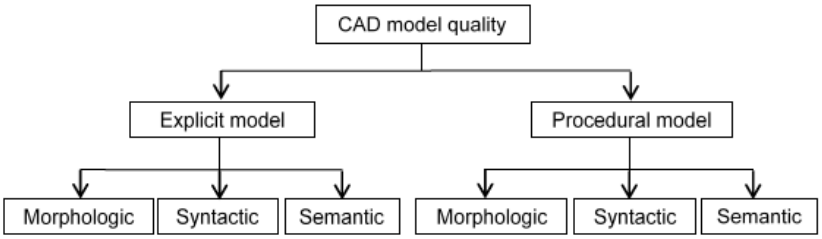


Figure 4. General levels of CAD model quality [30].

Ballou et al. [30] use four dimensions to measure the data quality: accuracy, completeness, consistency and timeliness. Accuracy of the model can be measured by comparing the starting parameters for the design and the parameters used in the model. This approach is not optimal, because it does not differentiate the importance of parameters. The aim should still always be to create the model using the exact parameters to create an accurate model to the design intent. On a semantic level this is very hard to verify and the responsibility is mostly left for the designer. [30]

Completeness defines that the model has all the data, components and geometries for the complete model. It may also include other type of requirements such as having full history, design intent notes or other engineering remarks. Completeness can be somewhat forced by forms and design software themselves. However, in the long run the requirement usually evolve, creating uncomplete models in the process which must be updated. Completeness can be measured as number of needed field filled or for example by counting the number of components. Consistency means that the all the data/models should be consistent from their form. This means that they have been created using same type of rules, same methodology and even with same tools. Consistency also means that different engineering system sharing the same product data should contain consistent and synchronized date. Timeliness reflect how up-to-date the current model is. Furthermore, having timeliness across engineering platforms results in having synchronized data. It is practically impossible to have all the data points or models on hundred percent timeliness but the aim should be to have as high percentage as possible. [29][30]

Company et al [31] use a division of six dimensions of quality in CAD models: validity, completeness, consistency, conciseness, simplicity, and conveying design intent. This has a lot of similarities to division done by of Ballou et al but also has some additional aspects. Validy in this case means that the model can be retrieved and be used, or simply put the model is available for use. [31][32]

Table 1. A list of data quality dimensions identified by company [33].

1	Free of error
2	Objectivity
3	Reputation
4	Believability
5	Relevancy
6	Value added
7	Timeliness
8	Completeness
9	Appropriate amount of information
10	Interpretability
11	Understandability
12	Consistent representation
13	Concise representation
14	Ease of manipulation
15	Accessibility
16	Security

Total Data Quality Management identifies several more dimensions for quality that are presented in table 1. This includes reputation, believability, interpretability and security to name few significant ones. This dimension listing also acknowledges more abstract properties of the model to promote high quality data. Some of the dimensions in the list such as believability are derivable from the success of overall model quality and thus are not well suited from verification purposes alone. However, these are good indicators for what is the quality level in the system currently. Additionally, some sources add reachability and accessibility to the list of quality dimensions. [33]

3.4 Error sources and types

Errors in product model can originate for example from bad working practices, lacking user skills and built-in from the methodology. User action related issues can stem from either user techniques or simply from mistakes during design work. It is impossible to have zero errors resulted from the user which highlights the importance of model verification. Errors might also occur because of the syntax of the database itself. This can be a result of bad housekeeping in database or even read/write errors on the storage device. General data errors include missing data, wrong data type, dangling data, wrong categorial data, outdated data, inconsistent spatial data, name conflicts and structural conflicts. The database errors might also be errors in simply reading or writing data. Problems may also stem from as early as part design and manufacturing requirements. These problems are also almost impossible to remove from the design process, as some changes might be necessary to create a complete product. Lack of time in design process also influences the quality as hurry often results in more errors and less time for verification. This is very common in current day engineering work and is an important aspect for the management to consider in the design process. It is common to overlooks the benefits of good quality and focus on minimizing the hours spent for the project. [6][34]

External contributing factors also exist such as program, database, conversion errors and technology limitations. Software and technology related errors and be diminished but some of them might be impossible to eliminate. CAD application algorithms can also result in some errors. Error caused by migrating data between system is also common in system with multiple engineering tools, especially in the data is not linked correctly. Poor change management in the product data management system might also be partly responsible for errors

in the models. Furthermore, different type of errors can also be identified. Contero [28] divides CAD errors to morphologic, syntactic and semantic errors. Morphologic errors mostly contain the geometrical errors such as topologically invalid models. For CAD and mesh models this includes errors such as tiny faces, narrow regions, non-tangent faces, narrow steps and sharp face angles. More geometrical error types have been defined for example in a standard ISO/PAS 26183:2006 that is used in automotive industries for defining suitable geometrical quality for CAD models. Product model errors also happen outside of the native modeling environment. The use of different environments and conversions can also result in errors in CAD models. [34]

3.5 Influencing product model quality

Product model quality can be influenced in ways such as preventing, checking and repairing. Preventing includes training of the modelers, creating and maintaining the modeling methodology. This might be the most cost effective method of increasing the model quality but in order to gain the benefits, methodology must be implemented in a coordinated and well described manor. Preventing requires a lot of resources, mainly in experienced engineer creating and spreading the knowledge of the modeling guidelines. In the elevator industry, the product segments vary from small, heavily electrical designs hall lanterns to big and complex, mostly mechanical assemblies such as car slings. This complicates the process of creating ubiquitous guidelines and uniforming the modeling conventions. Training the designers to use the methodology is also an important aspect in the quality process. This has been recognized as a sore point and improvements in this field could also yield significant rise in the model quality. [9][35]

Checking can be described as the verification and validation process of the models. Tools can be used for checking the models but some expert knowledge is still needed for the final checks. It can be automated to some extent, and so-called data gatekeepers can be used to ensure that the final model data will fit the common guidelines. Checking of the models should be conducted as early as possible to catch the errors earlier and maintaining the possibility to make changes. Fixing errors is the last possibility of creating better quality models. It can be very time consuming and create problems with allocating resources. Furthermore, it is not always even possible to make changes in some models because of inflexible model structure or the for protecting existing model data. [35][36]

Furthermore, SAS[35] identifies categorization, standardization and matching as three critical steps when improving product data quality. Categorizing data helps improving product data quality by enabling effective filtering of data and help identifying problems in certain categories. Standardization refers to creating rules and complying these rules for consistent data. Modularity can be built in for the data in standardization for supporting of easy interchangeable data models. Matching often uses the standardization for preparing the input. In this process, existing data structures are complied to the standards and duplicates can be eliminated via data analysis. [35]

Some future possibilities for influencing the model quality are semantic understanding, rapid learning, integrated governance and artificial intelligence platform. For example, AI systems could be used to check and repair the product data and possibly even follow the modeling guidelines.

3.6 Verification

3.6.1 Verification and validation

Verification has a direct influence on the product performance and product functionality. As terms, verification and validation are often used interchangeably. AIAA [37] describes model verification as the process of determining that a model implementation accurately represents the developer's conceptual description of the model and the solution to the model. In comparison, model validation is described as the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model. This difference is also illustrated in figure 5 that presents the verification being done mainly between conceptual and computerized model and validation done between computerized model and the real-life counterpart. [37][38]

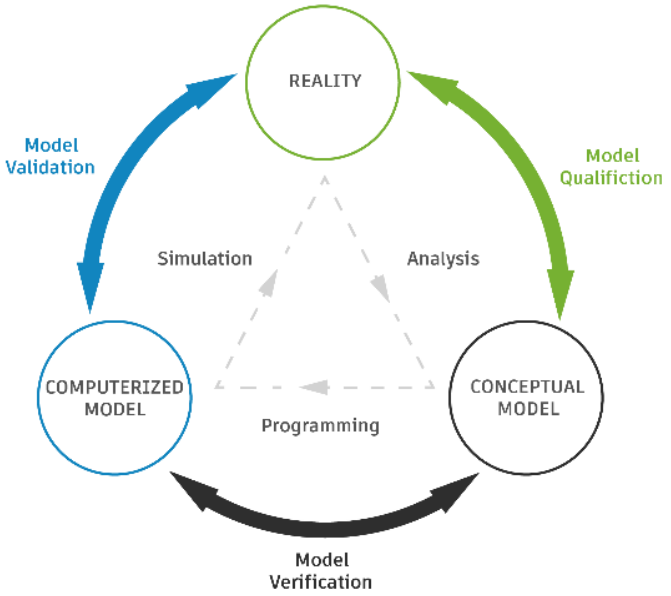


Figure 5. Model verification and validation [39].

The goal of verification is to collect evidence of the correctness and accuracy of the model for a specific scenario. AIAA suggest that verification and validation in fact cannot prove the models correctness and accuracy for all possible conditions and application but to provide evidence that the model is sufficiently accurate. The sufficiency naturally depends on the uses cases, but when that level for a specific application or situation is reached verification and validation process can be considered complete. [37][38]

Verification work is focused on identifying and removing errors in the model by numerical, analytical or even benchmark solutions. Validation however is about quantifying the accuracy of the model for example by comparing numerical solutions to experimental data. In other words, verification involves the mathematics associated with the model and validation focuses on the physics associated with the model. Thacker et al. [38] suggests that software V&V is fundamentally different from model V&V. Software V&V is required when a computer program or code is the end product and model V&V is required when the end product is a predictive model. A model in this setting is described as the conceptual, mathematical or numerical description of a specific physical scenario, including geometrical, material, initial, and boundary data. There are also differences between software, theoretical and data models. Verification can also be divided to code verification and calculation verification. In

code verification, the problems are constructed to verify code correctness, robustness and specific code algorithms. In calculation verification, the model is exercised to demonstrate that the model is computing a sufficient accurate solution. [29][[37][38]

3.6.2 Verifications role in model quality

As was discussed in chapter 3.5, the main players in model quality are preventing, checking and repairing. There are no clear figures on what kind of segments each of them hold but these different ways of improving quality need to be used together in order to have substantial effects. Even with extremely strict modeling guidelines some error can and will always happen and they can be identified in the verification and repair in order to improve model quality. Presented in first time in chapter 3.1, in the virtuous cycle of managing data quality the first action was define. Defining the metrics and the checking/verification parameters is done is mostly done in the modeling methodology. Modeling guidelines are created to prevent problems but also to define what problems exactly are by presenting a correct way of modeling. Verification can be harnesses to take advantage of the modeling methodology and to catch errors and design mistakes during the development phase. [40]

Creating a better quality on the first time the product is modeled will also bring benefits in the future. Verification can make sure that the model is easily modifiable and configurable and doesn't have any error that are hard or even impossible to fix after releasing, such as circular references. Verification is a great opportunity to limit the amount of rework done in the design process. As seen in figure 6, the amount of rework depends heavily on what point the design release is happening. When the errors and design flaws are noticed already before design release the amount of rework is small compared to noticing errors after releasing. [40]

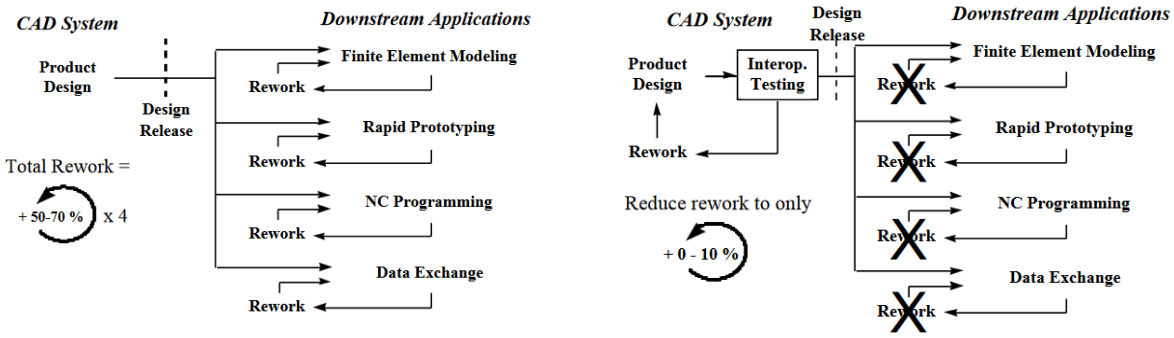


Figure 6. Comparison of rework with and without testing the product design[41].

Another important role of verification in product modeling is to work as a gatekeeper. By using verification metrics as a baseline for being able to submit the model work to the common database, the data has a certain floor for the model quality. This will create benefits especially in the long run, when data has been introduced to the standard and the model database has been slowly made to fit this quality. If no verification procedures have been established, the quality in the system can easily decay and defect are left unnoticed. [6][40]

Verification also plays an important role to achieving model quality adequate for advanced functionalities such as design automation. Verification can be used for determining if the modeling methodology is used correctly and for finding design issues and errors in the

model. In case of system that create models automatically based on parameter data it is crucial to have a robust and accurately configurable model to begin with. For humans, it is difficult to find mistakes and small error we have made ourselves. When combining this with huge product models with complex structures, verification done by hand would be very time consuming. [6][40]

3.7 Theoretical background of verification

3.7.1 Model checking and verification in general

As a term, model checking often referred to as checking of a mathematical model of a real-life system or object. This is not exactly the same as checking 3D-models or PDM models but they carry out similarities. Verification methods differ greatly, starting from the type of model. Different model types include system model, program models, geometrical models and data models to name a few. Methods used with geometrical and data verification are most important in the scope of this thesis but other methods will also be researched.

Yang [34] defines model as a conceptual, mathematical or numerical description of a specific physical scenario, including geometrical, material, initial and boundary data. Multiple methods have been developed for identifying issues in CAD model geometry. Modern CAD programs usually do this type of model geometry checks automatically and even doesn't allow the creation of such features. This enables the designer to focus on following the methodology and using the parameters that were initially given. Model checking can also be divided to different subsections such as explicit state, symbolic, refinement or symbolic trajectory model checking. Model checking can be performed on asynchronous systems as well. [34]

3.7.2 Verification methods

Formal verification is an often used method with design, mathematical and system model verification. Denman defines formal verification as the act of proving or disproving the correctness of algorithms underlying a system with respect to a certain formal specification or property, using formal methods of mathematics [42]. The models usually need to be simplified for formal verification. The model is a set of interacting systems and each of the systems have a finite number of states. These states and the transition between the describe the model. Using different states, for example current configuration and successive configuration the model can be written as a function of the present state and inputs. [42][43]

If the model is simple enough to be derived into discrete functions, rule-based formal verification such as binary decision diagrams (BDD) can be used. Rule-based systems apply rules, constraints or conditions a design and output values such as pass, fail or warning. In BDD, the model is broken down to binary decisions that can be evaluated using a set of rules. Its extension MDD (multi-valued decision diagram) is a data structure that represents finite valued discrete functions with multiple values. BDD and MDD require good ordering of functions input variables such as in inputs, outputs and state to efficiently represent the discrete function. [43]

A 3D model quality tool Q-raider uses design history for model verification. Design history refers to the chronological order in which features are created and to the constraints of overcoming limitations in a parametric CAD software. Q-raider consists of different modules with different functionalities such as geometry checking, knowledge reasoning, topological naming and name matching. Generally, in history based checking the first step is to extract

the design history to see how the model has been created. External file and formats can be used such as XML, but some design tools can also output the design history in native format. For example, in the case of PTC Creo, the design history is saved in a trail file in the working directory. PTC ModelCHECK also uses this design history trail file to assess the model quality and properties. [44]

Statistical analysis can also be used for assessing the product model quality. Dantan et al. [45] present statistical tolerance analysis based on constraint satisfaction problems and Monte Carlo simulations that calculates the probability of the product being possible to assemble. The calculations are based on individual parts tolerances. This method could potentially be implemented for further verification purposes. However, this only applies to easily quantifiable properties of the model such as dimensions or mass data. [45]

In future, advanced possibilities such as using Artificial Intelligence (AI) and neural networks can be utilized more efficiently on model verification. AI and neural networks could be used to deeply understand the quality aspect of the model by learning from based data or previously verified models. The benefits of using AI is to have better and more methodology based that rule based way of verifying model quality. However, no real-life verification tools are not yet available commercially but some research is being done towards functional applications.

3.7.3 Model verification automation

Automated verification for model is at the moment mostly dependent on native environment verification tools. Generalization can be made for automatic verification but the functional solutions must be created case by case. However, this is also natural as different companies have very different types of product models, requiring customization for the verification tools and software anyway. For example, the syntactic level of quality depends heavily on in-house modeling conventions.

Furthermore, the information related to semantic quality level is often difficult to find [46]. Information related to the semantic quality level is hard to find, because the modeling methodologies that provide the criteria for semantic/pragmatic quality belong to the enterprise's know-how. It is related to the domain of Knowledge Management, so it has a strategic value, and remains hidden to public diffusion. In the following sections, a detailed vision of each quality level will be given. [46]

Automating computational dimensions, data and tasks can be done with algorithms. However, the complexity of big assemblies and the references in them makes it much more difficult in real scenarios. A set of rules is defined for the algorithms and it runs through the model to be checked. The algorithms are usually hidden under a user interface. The verification automation is dominated by tool developers that create custom tool for each software or more ubiquitous tools that can handle different formats from multiple source software. In a rule based system, automated rule checking can be defined as software that does not modify a design but rather assesses a design based on the configuration objects and their relations and attributes. Eastman [47] suggests that almost all efforts in automating rule checking to date have been applied to building code and accessibility criteria. These types of rule checking are required of all buildings constructed within a jurisdiction. Automated code reviews can potentially save significant time and cost, since code plan checking is often a costly bottleneck in the building delivery process. [46][47]

Perera [48] has constructed a method for automatic configuration verification on complex software and hardware systems such as space mission ground systems. Configuration verification here means checking if the configuration reflected in the configuration management system, such as PDM concurs with what is provided. The use of configuration item data list from the configuration management system enables comparison between the listed configuration items and the reality. Filtering and autodiscovery of effects can be used to determine if the configuration was indeed created as specified. [48]

3.8 Verification tools

3.8.1 Verification tools for CAD models

As verification has been identified as an important aspect in creating high quality models, multiple commercial tool have also been developed. Practically all the modern 3DCAD modeling software have a verification tool embedded to them. It is not cost effective or possible to build model checker for specific modeling software in-house without input from the modeling software company. These model quality tools are primarily aimed at preventing easily solvable low-semantic level mistakes and incoherencies. González-Lluch et al. suggest [29] that model quality tools are mostly aimed at homogenizing the vast amount of documents produced and shared by large OEM's and are primarily aimed at preventing easily solvable low-semantic level mistakes and incoherencies. [29]

A study by González-Lluch et al. [29] showcases and evaluates CAD model QA and testing tools. Table 2, presents some of the model quality testing tools and their capabilities. The table shows both tools embedded into CAD programs and some external tools. It is notable that most of the design checker tools are linked to a particular CAD software but the linked ones generally come with more functionalities. The functionalities and the quality aspect used in external software alter between different tools. Table 2 presents that some verification tools provide more as the native environment embedded verification tools. However, in other cases these tools cannot provide the full experience the native tool offers. Furthermore, the study suggests that the morphological level is reasonably well covered in explicit representations. The syntactic quality in explicit can be better improved by using efficient modeling methodology. The quality aspects of procedural representations have not yet been thoroughly considered by model quality tools. The study also presents some examples of model quality criteria used in tools. For example, in SolidWorks Design Checker, some of the notable criteria are undetectable constraints, missing design intent, problems with geometrical accuracy, overridden dimensions and units used in the model. These kinds of checks provide support for good quality models. [29][31]

Table 2. Levels of available checks with some of the model quality tools [29].

Model Quality Tool	Explicit			Procedural		
	Morphologic	Syntactic	Semantic	Morphologic	Syntactic	Semantic
CADfix (ITI)	***	***	***			
CADIQ (ITI)	***	**	***	**	**	**
3D TransVidia	**	***	***	***	**	***
Cax Quality Manager	*		*			
Design Checker		**	*			
DesignQA®	*	*	*			
GeometryQA®	**	*	**			
PrescientQA®	**	*	**			
iCHECK IT	**	*	***			
KnowledgeAdvisor			*			
KnowledgeExpert	*		*			
ModelCHECK	***		**	***		**
NX Check-Mate	**		***			
Q-Checker	***		***	***		***

External model verification and checking tools work primarily by two methods. Some tools use Application Programming Interfaces (API) to communicate and function with the modeling software. This verification tools developed for CATIA utilizes the native API for communication. Another method is to use the model files fully externally and read them in their original data format. The difficulty of external services comes when the external software can't use the native format. This makes it necessary to exchange the format potentially altering the data and creating or hiding model flaws. Even though there has been attempts to have a standardized neutral CAD format, the still isn't a definitive support for one format. These neutral formats also have their limitation and possibly can't convey all the design intent or model rules and calculation in the new file format. There are also other types of options to export product model information such as txt and xml files but preferably the data should be verifying in its native environment. External model quality tools are also always an additional cost. In general sense, external verification and model checking tools should be used only after the native checker is fully implemented and development option would be costlier that and external tool. It is also possible to use 3rd party services that verify the models. This usually require models to be sent in a native or neutral format for the service operators. The actual verification can be done automatically with software or even manually. [29][49]

In some cases, the CAD data needs to be converted to a different CAD format. These conversions may lead to error in the models, on geometry or data level. Thus, also exchange verification should be taken into consideration even though it does not belong to the main scope of this thesis. Exchanging the data between different CAD formats is a common contributor to model errors. Because of this, multiple tools have been developed for exchange verification of product models. They usually focus only on the geometrical data as transferring for example parametric and product rule information is not fully implemented as a standard between multiple formats. CAD conversions are widely used inside and between companies and subcontractors. [29][49]

3.8.2 Verification tools for data and product data

Research indicates that most of the tools used for verification in PDM software are system specific. This is probably because the PDM systems have different data structures and schemas. PDM software have tools for checking the data integrity and correctness. Most common are the simple data and syntax checks for example to verify that the field is not empty and the value is the correct format. Some PDM systems are capable of doing cross-platform checks through APIs to compare product data between systems. [50]

PDM software are also capable of conducting regulatory compliance checks. For example, Solidworks PDM professional system help companies to comply with government regulatory requirements or industry standards by ensuring the regulations are met. PDM systems can also be used to keep the data in the system consistent by implementing checks and checking the product data structures. PTC Windchill also supports the verification and validation in its native environment with data structure functionalities. Furthermore, "gatekeeper" functionality can be set up for the system to govern the quality of data coming into the system. [6][50]

Furthermore, tools for data quality also exist that can be utilized for product data. A study by Barateiro et al. [50] present multiple different data quality tools from commercial or research origins. The tools can be utilized for example for data transformation, duplicate elimination, data enrichment, data profiling and general data analysis. Most of the tools are not system bound tool and they can be customized. SAS whitepaper [36] suggests that most of the customization effort is teaching the tool how to understand what are essentially the vocabulary, spelling and grammar of the product data "language". Data quality tools can aid by helping search for key words, phrases and other logic needed for categorization. Additionally, they provide methods to rank their matches and search results with numeric probabilities, weighed percentages or confidence levels and even potential matches that require manual review. [35] [50]

3.9 Industry benchmarks for model quality and verification

While the management of the company's design quality is usually quite critical information and thus not public information, some examples of using verification tools can be found. First example is from NASA Los Alamos National laboratory. The development center uses 3D modeling for shuttle related design and was not satisfied with the model quality. Using Pro/Engineer, the laboratory chose to use the included ModelCHECK tool for verification of the design work. Some quality problems identified were the use of external or circle references, incomplete information or failed geometry checks. They also utilized a gatekeeper functionality to make sure that the design content that was signed in to the PDM system was up to par. Another example for using ModelCHECK is Skoda automotive. They also identified multiple problems with their CAD model creation. These include using wrong start models, bad features and references, missing parameters, coordinate systems and overall bad model structures. Along with ModelCHECK they chose to use a PLM specific verification tool Vectorworks Konsistenzcheck that evaluate the consistency of the internal data. [51][52]

Some use cases can be found also for external verification software such as CADIQ. For example, Ford utilizes CADIQ software to eliminate most of the downstream rework related

to design data. Others issues were deemed to be lack of effective product data interoperability and CAD model reuse. They used CADIQ Six Sigma tool for identifying CAD model quality defects. [53]

Gerace [49] conducted a survey in a thesis for the use of product model verification tools. Figure 7 present the survey results for a preferred method currently used for verifying model data. From 27 respondents 10 are using automated verification software, mainly for verifying physical attributes and geometrical properties. The most common verification procedures used in the survey was comparing archival data formats to original models. In a further question, a best verification process available was inquired. With ten respondents, 5 identified automated verification and validation software as the best option. [49]

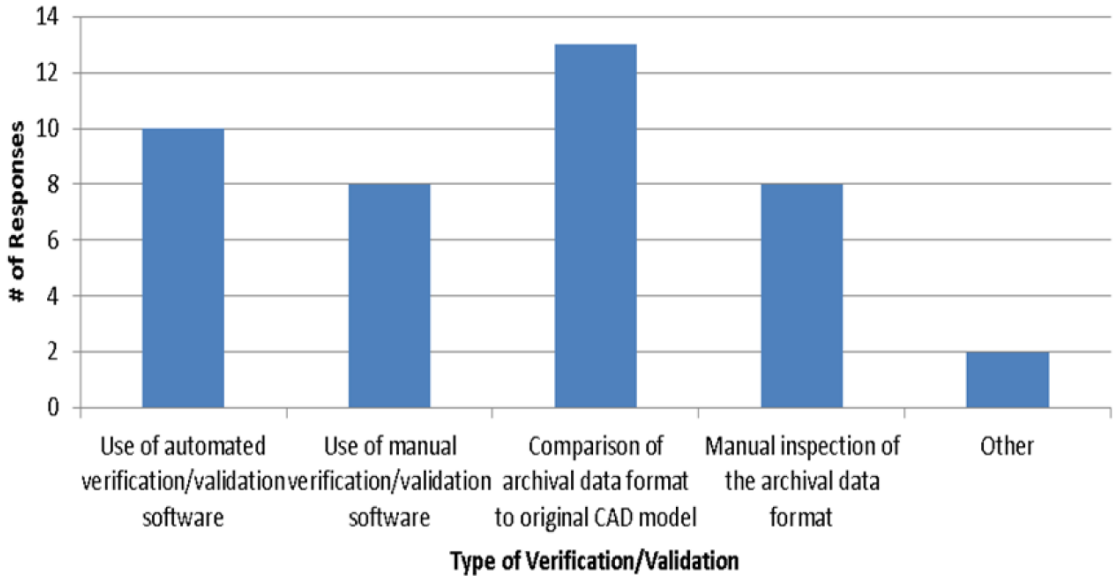


Figure 7. Use of verification tools [49].

4 Current product model verification process

This chapter is based on interviews and questionnaires done within KONE. Different products and units to get the largest representation. Answers were gathered from Finland and Italy. The purpose of the conducted survey was to get a general idea of the current status of model verification and model quality in different units.

The template of the questionnaire can be found in the appendix. However, the face-to-face interviews [54][55][56][57] deviated from this to some extent as time was limited and more important aspects for the interviewee were focused on more.

4.1 Current verification process

A lot of verification is still done with 2D-drawings and even though 3D will be implemented will be the main platform, 2D checks will remain a part of the process for years. The model verification is mainly done by the design engineers themselves when working on the product. This means that a lot of trust is in built-in quality and the skills of the engineers. Even though this showcases the trust in the work of the engineers, the system is prone to errors that are not easily identified by the designers. After the designers own checks, the model is forwarded to managers for acceptance. [54][55] [57]

Every product has a product manager appointed that is the final person responsible for the quality of their unit's product models. In C-process design, the designer has a bigger responsibility since making custom changes on the lower levels of the model can make it very difficult to verify by other personnel. The feedback for the models is mainly constructive comments from manager and senior engineer that have review the models. Portion of the components such as commercial library components are designed by subcontractor abroad. This makes the verification process more time consuming and heavily relies on written instructions and feedback. C-process design is not considered enough when creating the platform 3D models. This would be mutually beneficial since the re-usability and flexibility of the models would improve. Problem is allocation resources and shifting costs from c-process to platform development. [54][55][56]

4.1.1 Model quality

The interviewees are mostly satisfied with the model quality. One common aspect that could universally be improved is the mindset for following the guidelines. However, not all units are fully satisfied with the quality. One of the main concerns is that the re-usability of old component not on sufficient level. In some cases, model quality has been identified as a sore point in the process. From example, the use of some tools in the modeling software such as interference checks are not possible because of differences and errors in CAD models. Also, the modeling conventions themselves differ between units which make them incompatible in some cases. From C-process perspective, the modelling methodology used in different categories differs a lot, so the starting the model for them differs a lot in terms of usability. [54][55] [56][57]

A common wish is also to consider the quality more when creating new model. The repairing and fixing old models is lacking at the moment. Quality should be built-in, as in modelers should use the correct methods and follow the methodology. Furthermore, problems with legacy data has been identified as a point of emphasis. Another common concern is the time and resources allocated for design and model verification and repair. [54][56][57]

4.1.2 Common verification practices

So called SO-documents exist for steering the rules and structures inside the platform. These documents are usually kept quite uniform on a category level, such as between different sling products. Guideline documents for general modeling and for special component groups exist but for verification only a generic level document exist that suggests running test cases and designer knowledge to verify the model and the structures. It is normal for the designer to verify the model while modelling. Although this is an effective for time consumption, it is not an optimal practice because designer is trying to catch his/her own mistakes. Even if second designer or platform manager checks the model, when handling large models, mistakes and design errors can easily go through the checks. C-process also includes a verification round that is done by peer-review. In a case of highly modified structures this may not be enough to find all the errors. However, the requirements for c-process model is somewhat lower as they are only fit for single use cases. [54][55][56]

Configurability and robustness of models is verified by running multiple test cases in batch files, and then assessing the results. Currently there are no unified guidelines on how to pick the test cases and designer is guided to use "a suitable amount" of test cases. Visual inspection is also done for single configurations. Also, constraints and the use of correct parameters should be verified at this point. For full configurability, the rules and calculation scripts need to be verifying and tested. This also include checking the syntax of the scripts. Once again, batch files are used to test different configuration scenarios in effort to find any errors. On PDM, the most common verification done by the designers are checking the position number exist and are correct. This can be done automatically via release validator by comparing it to 3D model data. Release validator is a powerful tool in the PDM system that can be used to verify data consistency between engineering systems. However, its potential is not fully utilized due to concerns for effect to existing products. [54][55][57]

4.1.3 Common needs

Even if the product segments differ a lot, there are some common needs. Verifying configurability and re-usability are seen as important areas to development in the near future. Managing references is not always on par with guidelines or design intent. This weakens the product quality and is a major factor in poor configurability and makes making changes more difficult. This also goes further for having models design with C-process changes in mind too. Using for example skeleton templates for certain product categories could help with the modeling of product that share same type of geometry and key characteristics. Furthermore, especially the standard and semi-standard components should possess high quality as they are used in multiple assemblies and thus have larger effect of overall model quality. [54][55][56][57]

Making configuration verification easier would also be one universally beneficial aspect. A method for choosing correct test cases and the number of them would help standardizing the product model configuration verification. Verification tools for being developed specifically for configurations and this aspect should be taken into consideration in the future.
In current model structure verification, an overall view of the model structure would be beneficial to get the big picture while manually checking the model. This could include for example overlapping product information for structure and parameters. A tool that somewhat addresses this issue is also on the way. [54][55][57]

There is also need for more collaboration between the product segments and the c-process design. This should include sharing the modeling methodology, communication design intent and creating product models good for both a- and c-process cases. Consistent data is needed throughout the modeling process for example in data flow from R&D to production and service. Implementing the modeling guidelines on a product level is deemed be challenging. Having a more effective way of using and monitoring the use of common guidelines would unify the modeling conventions. Additionally, data and information flow in the system should also be increased and improved. Checking of the 2D documents such as configurable drawings is also a common need and should be a development target in the future. Some type of quick check would be optimal to catch most glaring errors in drawings. Publishing 2D drawing takes significant time with large assemblies and catching errors before this step would be extremely beneficial. This is true in both A- and C-process. [54][55] [56][57]

Another common need is the possibility of using additional tool in software such as interference checks and mass calculations. The current overall model quality is not on par to use these tools for reliable level. For example, some standard components interfere with assembly components, creating a large amount of errors that could have been avoided by proper modeling of one single part. Some model repairing work is needed to fix these types of issues. Some type gatekeeper functionality or stricter verification before release would be good to have in place to prevent low quality models to be released in Windchill. The need for making sure the models are re-usable and modifiable before releasing the product for production purposes exist on all the products. This could also lead to better overall changeability of the models and steer design to make any big changes before releasing it. [54][55][57]

4.2 Verification tools in current system

4.2.1 3DCAD environment

PTC Creo native verification tool ModelCHECK has pre-set verification configurations for different scenarios. Creo also includes a built-in geometry checking tool, that can verify that the models are compliant with VDA standard. Other internal Creo calculation and analysis tools are sometimes used if the models are on a proper level. The software even features a support for creating custom analyses and external Mathcad analyses.

Older version of Windchill had a release validator function that checked some of parameters and the product structure but in the newer version this has been taken as the standalone PDM has this check available. Currently, Windchill is not used for verification purposes in the company.

CAD configuration verification is either done on a local Creo session or using CAD robot. The designer uses XML configuration forms created by PDM system or creates the different test cases by hand in using spreadsheet and import them to Creo. It is possible to run a number of different CAD number configuration by using Creo batch functionality. Alternatively, a CAD robot can be used. CAD Robot automatically creates 3D models with parameters given in XML files. Technically speaking, the CAD robot runs the same functionalities but has the advantage of doing it remotely leaving the designers computer resources for more important use. The usage of CAD robots is increasing as more products are created suitable for the platform.

ModelCHECK is currently not fully utilized. The checking parameters are not optimized and different checks are not yet implemented to their full potential. No real guidelines for utilizing the tool so the responsibility and interest of using them relays on the designers themselves. Based on interviews in the company some designer utilize the different Creo tools on a daily basis and some never use them. Making these tools low-effort and easily understandable can help creating better quality model and to let designers focus on other design aspect on model.

4.2.2 PDM environment

The current PDM environment has multiple verification tools that can be used during the design of the PDM model. One of the most important tools is the Release validator that is used while moving the product state onward from design phase or when manually synchronizing data with Windchill system. Release validator compares the Windchill PDM data to the VariPDM data and reports missing or conflicting data. This functionality is carried out every time a synchronization between PDM and Windchill environments is done. The functionality can also be run manually by the user. PDM/3DCAD release validator is in common use. However, this tool could be utilized more to catch more errors in product data. The problem in more thorough implementation thus far has been resistance from the design field as this slows down work in crucial waypoints of the projects. Different type of reports and Bill-of-Materials can be exported from the PDM system for external analysis. Comparing of certain product structures can also be done in the current PDM system.

Configurable product model structures contain a lot of rules and calculation scripts for parameter. These rule and calculation can be verified to be formed in a valid syntax to prevent failures. Sometimes product model attributes need to be downstreamed in a product structure for configurability. For this purpose, PDM has a multilevel attribute assignment checker tool that can be used to verify that product passes on the correct attribute between multiple level in the assembly structure.

Verification of the models configurability is also done by running batches of file to test it with multiple parameters to find errors. A separate functionality exists in the PDM software for this. The correctness of configuration is verified automatically while creating the configuration. This verification checks that the needed parameters are functional and that the rules and calculations do not create errors during the creation of the configuration. The PDM configuration is also commonly run in the CAD program to verify that the structure and the model is visually correct and does not create any errors there.

4.2.3 Configuration verification and validation suite (CWS)

Configuration verification and validation suite (CWS) is a program interface specifically developed for verifying product configurations. It uses PDM structures for configurations and runs these different configurators on CAD robot server to created configured 3D models. CWS outputs the configuration in a user interface which enables the user the view a 3D viewable model. 3D viewable models are lighter version of the 3D model that are adequate for visual inspection. CWS also outputs if the configuration is successful and shows occurred errors. It uses user given parameters to configure the product model in PDM, then send the PDM configuration to CAD Robot to create a configuration specific 3D model. After CAD robot has generated the configured 3D model, it provides the user with a 3D viewable model.

New functionalities are being develop to improve the tool and for example outcome of this thesis is a subject to be used as a CWS functionality.

CWS is currently in the development and testing phase and new features are planned to be added. The future goal is to use CWS for the configuration V&V throughout the company. By adding and improving functionality on the CWS, a widespread use will be probable in the future.

5 Development of product model verification metrics

5.1 Product model verification dimensions and metrics

Pipino et al. [9] suggests that a "one size fits all" set of metrics is not a solution and that assessing data quality is an on-going effort that requires awareness of the fundamental principles underlying the development of subjective and objective data quality metrics. Based on the literature research, key metrics for product model quality for KONE were identified. The main dimensions for the metrics can be seen in figure 8. The dimensions contain almost universally agreed model quality dimensions of accuracy, completeness, consistency and timeliness. These dimensions should be the basic principles when assessing product model quality. Additional often used quality dimensions is accessibility or reachability of data. However, for all-around measurement for quality of configurable product models, even more additional dimensions are needed.

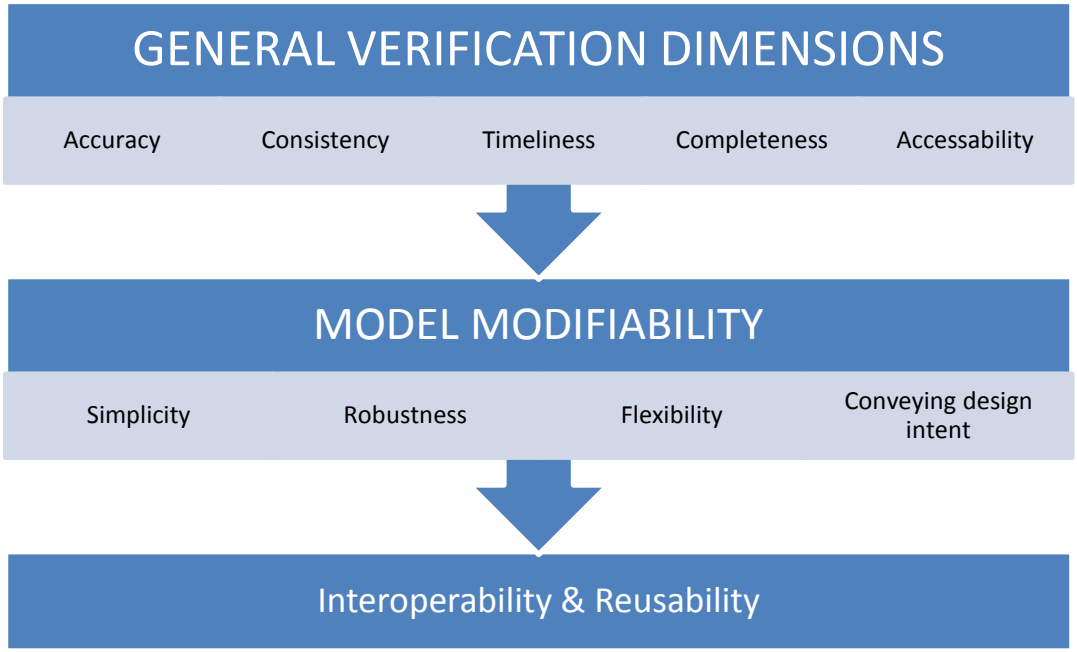


Figure 8. Model verification metrics.

Figure 8 provides an overlook in the dimensions chosen to be used in this thesis. It differentiates general, modifiability and re-usability aspects into their own sections to help categorize effects of a certain dimension. Categorizing will help evaluate what needs to be targeted to improve product model quality on a certain dimension.

5.1.1 Metrics for general verification dimensions

Model accuracy one of the most important aspect of model quality. Accuracy can be quantified for example by measuring the difference between starting parameters and the parameters used. In reality, this is difficult to measure since the starting parameter data is usually not formatted in a way that comparing them would be easy or even possible. The accuracy aspect depends heavily on the verification done by the original designer as the accuracy is difficult to verify on a semantic level. The most important aspects in accuracy is that is the data correct overall, is it as intended and is it formatted correctly. Following the guidelines is a needed overall aspect in model accuracy. Accuracy is by default needed when creating high quality models. [9][31]

Figure 9. General verification dimensions.

The completeness of the model is described in multiple metrics. In general, the model should contain all the data, components and geometries needed for the complete model. Requirements for this vary widely between different product sectors and the responsibility again depends heavily on the designer. Some completeness aspects of the model can be driven from the modeling environment for example by having compulsory data fields. 3D modeling software also requires a certain level of constrains and rules to create the geometries which directs into making complete structures and assemblies. However, also the completeness verification still needs to be done mostly by the designers and other people responsible for quality. For example, it is technically possible to check the presence of components in certain positions in BOM, but without designer's knowledge it is practically impossible to determine if this is empty position is an error or an intended change. [9] [31]

Timeliness can be measured directly by using the timestamps of modifications or creation in the system. However, it is extremely difficult to keep track of changes in a system with thousands of models. One method is to utilize cross-system information in the design environment and to compare if other systems contain the same data. If the data is different, it is probable that one of the systems does not have a timely model. As is the case with other quality dimensions, the model timeliness is heavily relying on the designer and other people responsible of the model or product. Out-dated product can potentially be identifying by taking note of the last update date of the model. [9] [31]

Consistency can be verified by comparing the data and the models to existing rules and methodology. For example, it might be that the model need to have a universal structure to operate optimally in the full design environment. It also means that the data should be matching the data in parallel systems. This can be verified for example by comparing data in a chosen format. Furthermore, consistency can also be described as a syntactic dimension of the model and that consistency creates syntactic quality. Company et al. [31] identify multiple metrics for estimating the modeling conventions for syntactic verification. Measuring the effects of the modeling convention directly is difficult but verifying some of them already is or can be automated. Syntactical level of model quality comes from using the same modeling conventions and methods. On a lower level this can be verified by using correct starting templates but still most of the effect on syntactical quality comes from designers themselves. Semantic quality of the model transfer to the re-use and modification abilities of the model. [9] [31]

Product models also need to be accessible by the designers, managers and other personnel that need the data. Wang [33] defines accessibility as the extent to which data is available or easily and quickly retrievable. Using too strict user control settings can degrade the re-use of models, prevent designers getting accurate information and simply to loss of time because of access requests. [33]

5.1.2 Metrics for modifiability dimensions

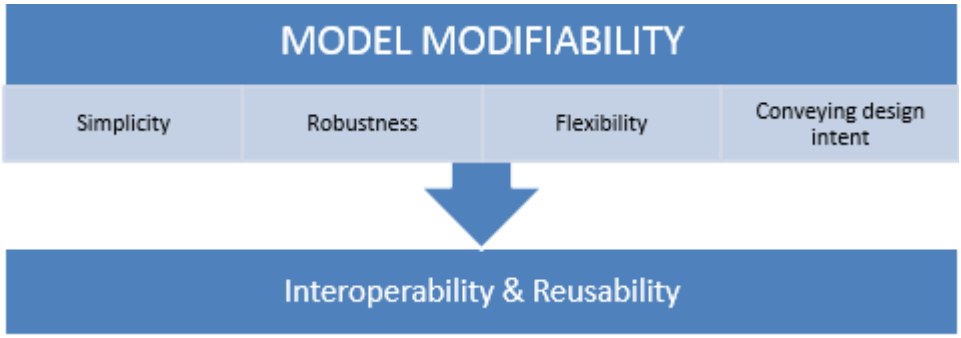


Figure 10. Model modifiability dimensions.

Simplicity of product models can support other quality dimensions. For example, simple models can promote consistency in the modeling system. Using simple references and constraints can be helpful in eliminating configurability issues and it can make the model more robust overall. However, simplicity is not the holy grail of modeling. As Rodriguez-Toro et al. [58] present, adding simple features or models together may not result in a simple model anymore. Simplicity can also be described as product model structure being understandable. [58]

Robustness can be defined as character that model is resistant to error while being modified. It is a good indicator for overall model quality as it is a result of minimizing error and quality defects from models. Few aspects of creating robust models is to create simple models by using simple features. Also referencing is important for robustness in configurable models. Currently, robustness is mostly being tested in CWS as running batches of models and verifying the results. However, further notice should be given already in the modeling phase to ensure that the referencing and other important aspects are according to parameters. Especially reference have a huge impact of the robustness of the models, as most of the generation errors are related to bad or insufficient referencing. [32]

Flexibility can be described as a measure for the range of reachable states and the time and cost required to change state [59]. Flexible systems are built for a set of reachable states which are predefined during the engineering process. Interoperability describes how a master model can be accurately transferred from one modeler to another. Efficient way to promote interoperability is to create and follow common methodology for modeling in all systems. Good level of interoperability brings benefits especially in cases where product model data is widely used for downstream applications. [59]

Reusability is derivative of the structures and references of the model. In model reuse, the master model is modified in the native modeling environment to be utilized in other situations. Reuse can be performed at different levels from utilizing library component to using existing designs of similar properties. Two primary methods for re-use are instancing and cannibalizing. Instancing refers to redesigned versions of the master object and cannibalizing refers to using old designs in newer models. In the case of reusing configurable product models, adequate and correctly formatted rules are needed. As the modification to the models in the case of re-use are frequent, the quality of reusable models must be higher than normal as they need to reliably allow for modifications while maintaining the original design intent. Design intent and rationale are always needed for model to be re-usable. [29][31] [60]

Conveying design intent is also a crucial part of configurable and re-usable models. Some newer engineering processes such as MBD and KBE are utilizing the design intent better than current widely used processes. Methods need to be provided to the designer to help convey the intent of the original design. These methods often are as simple as comment sections and notes. However, a clear recommendation should be set to help provide more complete information about the intent of the original design. Interoperability describes the level in which the product model data can be transferred into a different format. Interoperability is a crucial quality aspect for models in engineering systems consisting of multiple design environments such as CAD, PDM and CAM. It directly effects the quality on the models on the downstream applications. [31]

5.2 CAD specific metrics

On a morphological level the model quality can be quite effectively quantified even with CAD environment native tools. Different type of geometry checks and identification of topological errors is usually even built-in the software to make sure the software works as it should. For example in Creo, missing references inside a sketch produce error messages and prevent the user from creating geometrically impossible features. Standards, such as ISO/VDA 4955 have been created for evaluating the topological quality of the model and these should be at least considered to be used in the checking procedure. Tables 3 presents examples of some geometry metrics used in standard VDA 4955. The metrics have certain values that they need to fulfill to be applient for the standard. Note, that there are also Boolean options such as for "Multi-body solids". González-Lluch et al [29] suggest that discrepancies between geometry and functionality result in semantic level errors in CAD models. Morphological quality is also large factor in better quality in downstream application models. [19][28][29]

Table 3. Geometry quality metrics in VDA 4955/2 [28].

Code	Description	Criteria
M2	Identical elements	> 0.02 mm
SU9	Min. curvature radius	> 0.5 mm
F15	Distance to its surface	< 0.02 mm
T18	No. of faces per edge	< 3
SO26	Multi-body solids	No
D28	IGES conform texts	Yes
C7	Distance to itself	> 0.02 mm

A model is simple if the model tree is clear and understandable. This means that the modeling operations in the modeling tree must be labeled to convey their functionand related modeling operations must be grouped to convey the parent– child relationships in the model. A model is also simple if it uses compatible and standard modeling operations. In fact, standard modeling commands have been proposed as mechanisms to exchange design intent [61]. Model should not contain any repetitive constrains, modeling operations or datums. [29][31]

Completeness of a 3D model must be identified by the designer themselves. Theoretically it is possible to use redundant references and suppressed features as an indicator for model completeness. However, only the designer really knows what need to be in the model and what does not even with design guidelines in the place. On a geometrical level B-rep, mesh or overall surface checks can be executed to ensure the geometry is complete. Geometry can

be considered complete if it replicates the size and shape of the part being modeled. Modeling software can identify geometry issues quite easily but while these problems might be possible to detect, it is often impossible to automatically solve them if the designer's intent is lost. [29][31]

Consistency is a method that is very inheritant of used modeling methodology and the management and quality control that ensures it. It can be defined through modeling guidelines. Using well-defined common methods for featuring and other modeling aspects will have a positive impact on model consistency. Furthermore, having well-crafted start files and templates in the modeling software helps to induce consistency in the models. Consistency checks can be executed for example by checking that the correct templates, starting parameters and correct units are used. For example, family table models might have a set of company specific parameters that are required for all the models. Currently checking the family table parameters is manual work and thus errors can be easily missed. In feature-based CAD programs, there should also exist guidelines for a well-structured model tree. The current guidelines in the company address this but the verification of model tree is only implemented manually. Consistency also applies to the alignment and orientation of the models. If possible product, components or industry standards should be used for the correct orientation. [62]

Parent-child relationships in model are prominent for re-usability, flexibility and overall modifiability. As the parts are seldom edited during the initial design process, parent-child relationships can be untested and thus even well-working models might break when modifying them. Parent-child relationships can also easily cause domino effects in the model by failing features that leads to other referenced features failure.

For robustness in CAD setting, reference control, limits for input and output values and well-built rules are the most important aspects. Reference control involves using skeleton structures for referencing and positioning parts in assembly, using planes and coordinate system as the main references in parts and not having dependencies to other models. Detecting reference systems and suitable datums for skeletonizing models can be challenging. However, some research in ongoing on this topic. For example, Aleixos et al [63] have studied using skeleton datum systems to improve the CAD programs capability for conceptual design. The main referencing features such as the model skeleton should be located in the beginning of the model tree. Overall good practice is to have all the references as early as possible in the model tree. For assembly purposes, having pre-set, standardized interfaces for assembly will help with the overall assembly robustness. However, it should be noted that sometimes the modification of assembly interfaces lead to instability with already assembled components in the system. Furthermore, modeling operations such as sketches need to be fully constrained for the model to be robust. Insufficient constrain often lead to failing or disproportioned features when the original feature is modified. [29][63]

Models should be obvious for all the designers but intuitive designs are not always intuitive for everyone. Conveying design intent can be improved by modeling methods, comments and naming. Model tree plays a major part in conveying design intent in feature-based modeling. Ideal model tree is like a user manual to edit the model and the design decisions should be traceable within the model tree. The modeling operations should not be fragmented or overlapped and they should be labeled for quicker identification of the operation. Furthermore, datum should be able to convey the skeleton of the model. This means that there should

not be redundant datums or references and the skeleton should be as simple as possible. Geometric constrains should be used to highlight functional relationship in sketches and between parts. For example, coincident constrains with corresponding features axis can be very effective in communicating the position of another feature. All the functional relationship must be communicated, whether by using geometrical constrains as indicators or by commenting the model rules. Also, naming of features and modeling operation will greatly help with conveying the design rationale and the order of model operations. Sometimes features can be a part of multiple modeling operations. In these cases, gathering modeling operations in correctly named groups will help to convey design intent and help simplify the model tree. Groups should form logical sections of the models. It is also important that the model itself is the medium of communication for the design intent. All the needed information for modification of the model should exist in the model and be understandable. [11][29][31]

For good interoperability, the semantic differences between similar product models in different representations should be minimal or at least identified. For interoperability purposes, CAD exchange verification tools mentioned in chapter 3.8.1 are very useful. These tools are specifically made for minimizing errors for representation or format exchange to address issues with interoperability. Syntactic errors can often be the source of interoperability problems appearing because the use of different data structures between the native model and the downstream applications. [29][59]

5.3 PDM specific metrics

Product data in PDM presents additional challenges as product can refer to many different types of things. Product can be raw material, semi-finished goods, spare parts or finished goods. Additionally, configurable product data structures further complicate the data items in PDM. Because of this, the consistency of data in the system and planning of the data structures is crucial. Furthermore, the consistency of data between different systems in the modeling environment is one of the most important aspects in modern multi-environment engineering systems. The complexity of product data can too demanding for traditional, pattern-based data quality approached. Russom [23] suggests that semantic-based approaches that can adapt and learn nuances of new product categories is required. With this approach, standardization, verification, matching and repurposing of product data is possible. [9][64]

The complexity of product data also means that it is almost impossible to conclude when the product data is completed. However, some indicators such as missing BOM position can be utilized for implementing low level checks for fixed structure items. The importance of data accuracy in PDM can be seen for example when creating delivery BOMs from engineering BOMs. The delivery BOM should contain all the items that the engineering BOM does and missing items can result in shipment of in complete goods. [6][9]

Many key performance indicators (KPI) have been established for product data. For example, Stark presents ten KPIs for product data that can be seen in table 4. These metrics can be used to determine the quality level of data sets but they are not as suited for evaluating and verifying the quality level of single product data instances. Furthermore, some metrics or parameter thresholds need to be used to correctly evaluate if product data is incomplete or not. This can be done case-by-case but it would be better to have clearer idea of the quality metrics involved. Furthermore, Stark suggests that schema level data problems can be avoided with an improved schema design and that they do not depend on the actual data contents. Furthermore, instance level data problems are related with the data contents and

thus cannot be avoided with a better definition of schema since the schema definition languages are not powerful enough to specify all the required data constraints. [6]

Table 4. KPIs for data quality [6].

% of product data that is duplicated	% of product data that is electronic
% of product data that is incorrect	% of product data that is not under change control
% of product data that is incomplete	% of product data that has been lost
% of product data that is never used	volume of data that is re-entered manually
% of product data that has no owner	number of different copies of same document

A common need in verification in the company was to have a better way of evaluating configurability of a product. Jinsong et al. [11] suggest that there are three aspects of knowledge for supporting the configuration process: component knowledge, product topological structure and configuration rules and constraints. Component knowledge represents all component types that can be used in the end solution. This means all components should be characterized by attribute sets. Product topological structure represents the interactions between design objects and controls the whole configuration process. Extra attention should be paid when defining the interactions between items in a configurable structure. Rules and constants are used as a precondition for the final configuration. Constraint set are crucial for configuration verification as invalid constraints may lead to incorrect configurations. For a robust model, extra care should be used when creating and testing product platform rules and calculation scripts. Usually the robustness errors are only found while testing the configurability of the model. A visual explanating of configuration knowledge can be seen in figure 11. [11]

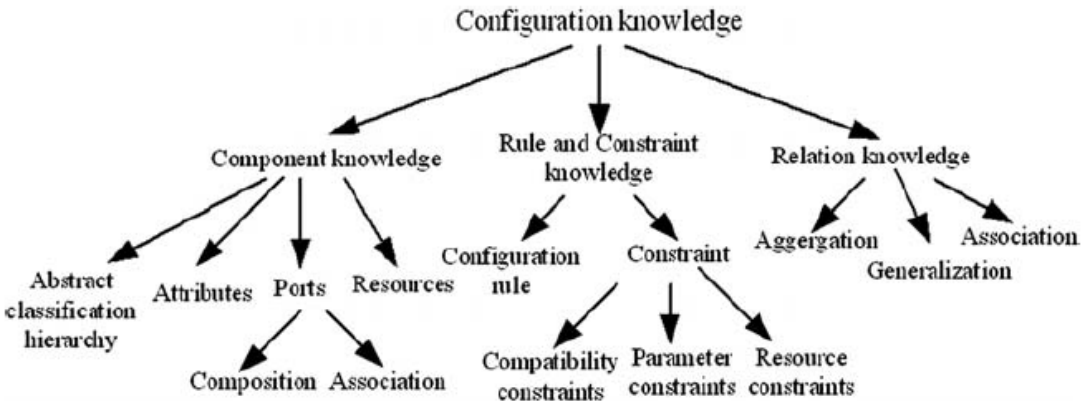


Figure 11. Configuration knowledge aspects according to Jinsong et al. [11].

6 Development of verification tool

6.1 Choosing the development platform

The model verification tool platform to be developed was chosen to be the PTC ModelCHECK tool included in the Creo software. It was chosen because it is already widely available inside the company but the its potential is not fully utilized. This makes it a good starting point for automation of model verification and can potentially be used as a testing and benchmarking platforms before bigger investments and changes to the modeling environment. MC also allows verifying the models in their native environment which should have a positive effect on minimizing error in computing during the verification.

Different type of model quality tools also demand different type of resources. Some of them on ready to go with minimal setup and some require expert knowledge of the modeling systems, models and the methodology. They also carry around different costs. As can be seen in the table by 5, the cost of an embedded software is significantly lower than other tools. This is one of the reasons why utilizing the existing tools should be a priority, instead of using external software. Furthermore, as per discussions with the company and my instructor, there is no desire to create completely new tool because of the maintenance and effort related into them. There is some development currently happening in the model verification field such as with CWS and the aim was to potentially provide something that could be implemented in the CWS tool later.

Table 5. Costs of different Model Quality tools [29].

Type	Average MQT Price one license (€)	Average Annual maintenance MQT (€)
Embedded and linked	1450	225
Not-embedded but linked	6900	1150
Neither embedded nor linked	15500	3500

Based on a brief industry benchmark research, ModelCHECK tool itself is used in organizations and companies such as NASA [51] and Skoda automotives [52] with good results. In the case on ModelCHECK, as the company already obtains the license for PTC Creo, the actual costs for using it as the verification tool come for launching and maintenance costs. Launch costs include creating documentation, guidelines and testing the MC. Still the total costs in wide implementation of MC are all most non-existent especially when factoring in the potential benefits in model quality instantly and in the future. Even though MC doesn't offer verification for all the metrics identified, it functions as ground work towards more complex product model verification tools and it can be used to evaluate the effect and benefits of using different verification tools. Creo API enables future development of fully specialized and customized tools for verification.

No PDM tools were chosen for development in this thesis for several reasons. The current PDM tools are developed by the software provider and the development possibilities differ a lot from open API system of PTC ModelCHECK and Creo. Furthermore, developing a new parallel PDM verification tool would have not been a desirable outcome for previously mentioned reasons. Product data in the current PDM system is largely dependent of CAD model data when the master model is deemed to be 3DCAD.

6.2 Choosing primary CAD verification metrics

The metrics used in this MC setting were deducted by going through the full parameter list of MC and evaluation each parameter towards the metrics and quality aspect found in section 5. For optimizing the different possibilities in verification, multiple different versions were created and tested as a purpose to address time consumption concerns and balancing between checking everything or only the essentials.

The main aspects to be checked based on the literature research and in-house interview were chosen to be constrain and reference control, configurability and changeability, conveying the designs intent, use of correct start files and parameters and geometrical quality. From these, the geometrical quality is more concerned as the indicator of model robustness as opposed to geometrical tolerances used.

Reference control focuses on where the references are done in the models. Preferably only datum and design skeletons are used for references instead of surfaces and other features. Configurability and changeability can be mostly assessed as the model robustness, referencing and feature control. Design intent is measures as if the model specific rules in the CAD system are commented. The use of correct start files, parameters and units can be verified directly. Also, the geometrical quality of the model is taken into consideration. Other aspect may also be included in the check for the best possible result.

6.3 PTC ModelCHECK in general

A set of text files are used for configuring the ModelCHECK. The files can be edited in a common text editor or in the user interface inside the Creo software. The file system consists of 5 main files config_init, set_conf, start, constant, checks (Figure 12). The config_init -file and set_conf- file specify the initial setting and determine which configuration file sets are read in the system. By using different config files there can be multiple sets of ModelCHECK configurations and they can be changed fairly easily. A help document [66] created by PTC provides the explanation and inner workings of the ModelCHECK parameters.

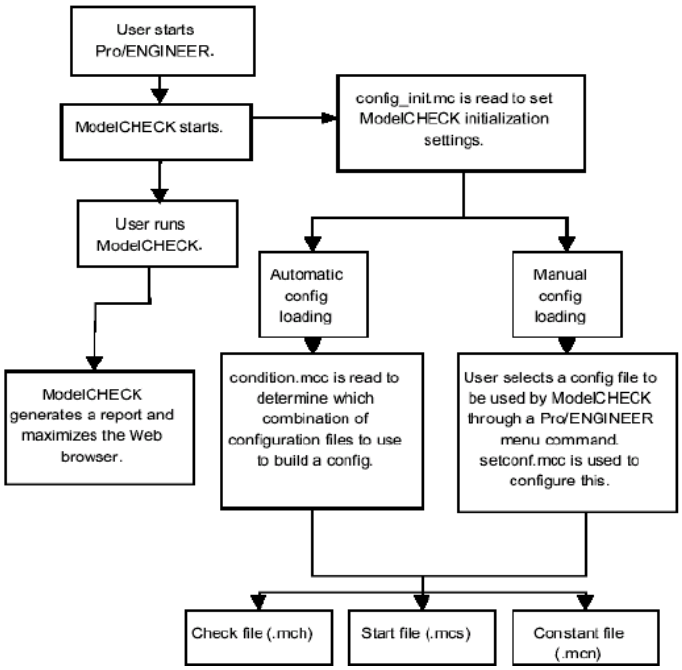


Figure 12. User actions and information flow in ModelCHECK file system [66].

The conditions file consists of IF-ELSE structures, parameters and operator. Technically, condition files can use any parameters that are used in the model. This also includes the data of the user doing the check, creation data, software version and model name. By using the "*" -character as wild card operator, also parts of the name can be identified. For example, "DEL_*" refers to all model starting with DEL_ prefix. The operator main consist of is equal to (EQ), not equal to (NEQ), greater than (GT), less than (LT), greater than or equal to (GTE) and smaller than or equal to (LTE). Different model types such as solids parts, sheets, skeletons and assemblies can be used in the conditions. For user control, different groups can be configured. This can be used for example to run stricter and more time-consuming tests by certain model checkers. Groups are identified in a text file with a list of usernames or set up by PTC ModelCHECK configuration tool.

The most important files for the checking metrics and parameters are the condition(.mcc), check configuration (mch), start parameter(mcs) and constants (mcn) files. Condition file specifies the exact ModelCHECK configuration files used for each scenario. The conditions can for example be used for running different checks for metric and imperial unit systems. Below is an example of how the conditions work.

IF (MODEL_UNIT EQ MM) config=(check/Kone_simple.mch)(start/Kone_defaults.mcs)(constant/mm.mcn)(status/sample_status.mcq)

This condition checks the model's system of unit parameter. If the model is modeled with millimeters as unit, ModelCHECK then select the following files for the configured checks. Conditions can also be used to override checks for some models. For example, condition

IF (PTC_WM_LIFECYCLE_STATE EQ Released) NOCHECK

checks the lifecycle state of the model and if the model is already released to production it will not be checked. This can be used to skip for example already retired model or models that are still in the development phase. However, as mentioned earlier, the benefits are bigger when the modeling errors are found in the early phases of the development cycle.

Start file can be used to set models standard parameters such as system of units and default names for coordinate system and planes. It can also be used to rename parameter names automatically to convert old parameter names to comply the specification given in the file. The statements in the file follow the pattern of parameter and value for it. For some star file parameters, the values are limited and some are free fields. Operators can be used further refining the parameter condition. For example:

*PRT_MODEL_NAME NEQ PRT**
PRT_UNITS_LENGTH MM

The first row checks the parameter for the part models name. The NEQ-operator indicates that part starting with PRT are not passing the ModelCHECK. This is used to prevent Creo's generic filenames such as PRT0001 out of Windchill commonspace.

Constant file defines thresholds for some of the geometrical check parameters. For example,

PERC_EARLY_ROUND 0.25

defines that the threshold for triggering a failed from using round early in the model tree is 25%.

There are four modes on how ModelCHECK tool can be run. Interactive check mode can be started by the user from the Creo control panel. This run the checks that has been configured for the this specific checks. Regenerate mode regenerates the model or models if in an assembly and checks the models based on the parameters defined for it in the check parameter file. Save mode check runs the ModelCHECK every time a save operation is done in the program. Because of this, the save check should be light to avoid time consuming checks with every save. The save check can also be used as a gatekeeper for the saving models in the PDM system. With batch mode, the ModelCHECK can be run outside of the Creo. A set of parts, assemblies or drawings can be defined to be run in batch mode.

The check parameter file configures all the different checks by listing check parameters that are used and identifying how the parameters behave in the check. They can have different values for different checks such as yes, no, warning. The start config file is used to specify the start part information for specific checks. There can be multiple mcs-files. Constant file is used to specify the constant values such as the length of a short edge. The check parameters are listed in the file and have the corresponding parameter values for each check type in columns. For example:

NUM_COMPONENTS YN Y Y Y N

This check parameter counts the number of components in assembly. First column after the parameter indicates possible values for the parameter. Next columns refer to each of the individual check modes starting from left Interactive, Save, Regeneration and Batch.

After the check is run the program a report opens on the screen. The report has a general status bar that shows the check status either in green, yellow or red. Thresholds for the number of errors and warning considered approved for each state can be modified in a status-file. The report presents the errors, warnings, notes and successful checks. If the checked model was an assembly, the user can also see a “check-BOM” that shows check results for all the individual parts or subassemblies. Error can also be highlighted in the model for better visibility. The report is in html format and it is usually saved in windows user’s temporary files or the working directory. The report output folder can also be changed in the configuration files. An example output of the report can be seen in figure 13.

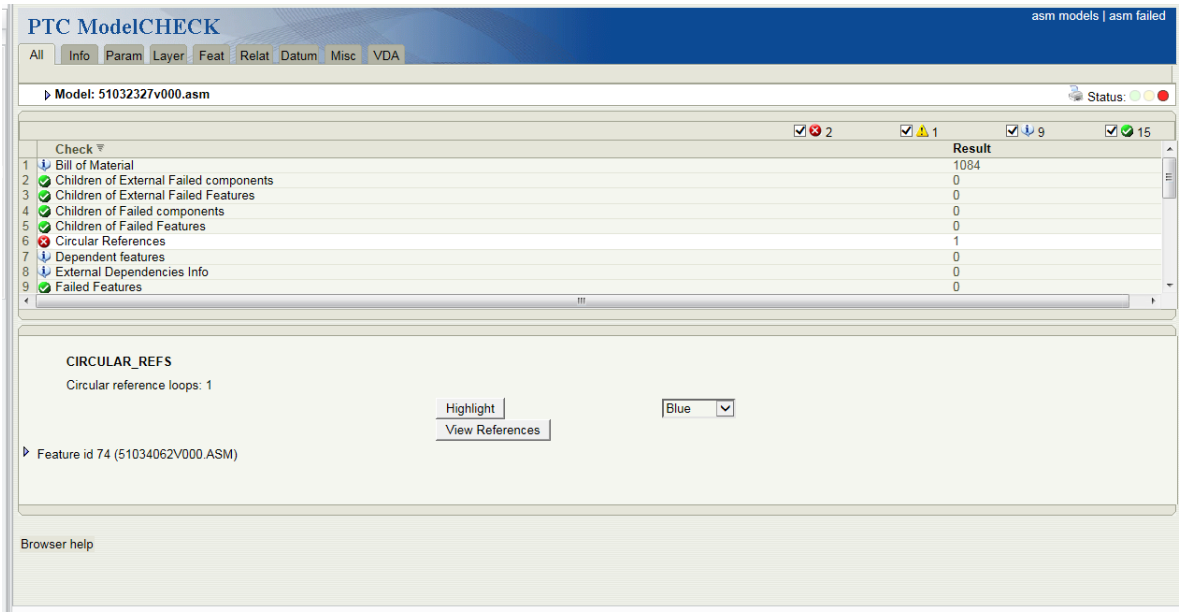


Figure 13. Example of a check report from ModelCHECK.

As can be seen in figure 14, the report also has different viewing options. in the top-right corner of the report fields four icons with select button can be seen. The first from the left (red button) refers to errors, the next is for warnings, third is for reports/lists and the final one on the right toggles approved checks on and off.

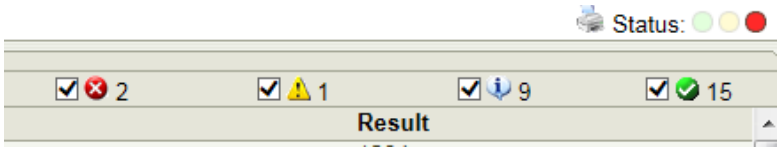


Figure 14. Different view options in ModelCHECK report.

6.4 Creating a custom PTC ModelCHECK

The ModelCHECK set up process is somewhat confusing with multiple configuration files and different setup files for different checks as can be seen in figure 12. On the positive side, even if the modification cannot be recommended for everyone, the files are available locally even for normal user. This also provides a possibility for designers to modify their ModelCHECK configuration files if needed. Completely new sets of checking files can also be created and saved parallel to the original files. Conditions or config files can be used to select these different checks. Setting up and modifying MC files can be done by an experienced user but deep knowledge of the company's design methodology is crucial.

However, the actual modification of the files themselves is easy. The files can be opened and modified easily in text editor and Creo 2.0 even has a graphical interface for ModelCHECK. In this case the raw text editor method was chosen because of simplicity and better overall picture. Programming skills not required for customizing the checks but understanding basic if-else structures will help when setting up the conditions file. Basically, modifying the checks and start files is just adding and altering parameters. As can be seen in the example file in figure 15, the check configuration file consists of a list of check parameters divided into part, assembly and drawing checks. These parameters have values in multiple columns that correspond to different check in MC. Parameters can be added to the list

and the individual checks are defined by the parameter value in the corresponding column. Explanation of all the parameters used in the checks created in this thesis can be found in the appendix.

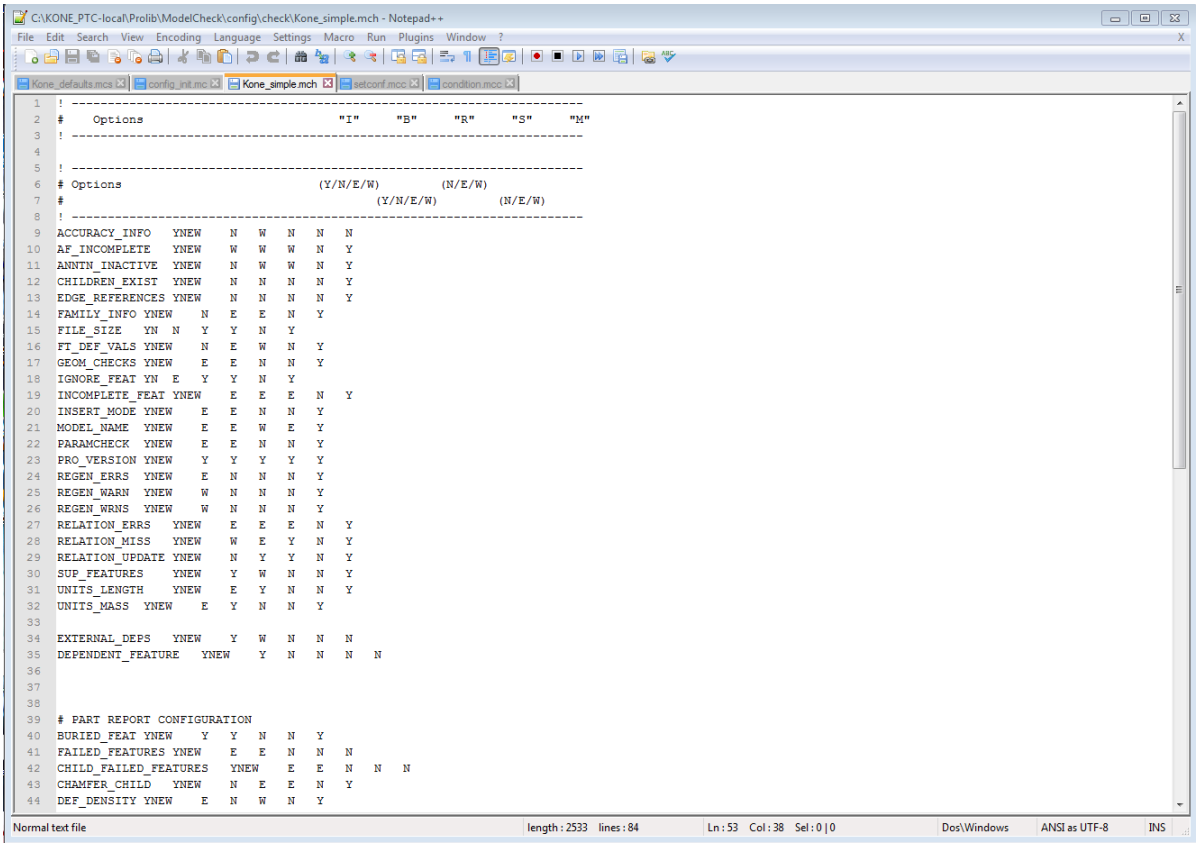


Figure 7. Modification of a custom check-file using text editor.

A set of old ModelCHECK files were used as a template for building the new checks. All the conditions and checking parameters in the files were explored and modified accordingly.

6.4.1 Conditions file (.mcs)

The conditions file has two constraints, on based on system of unit and one based on Windchill context location to identifying commercial standard components. The components from the commercial component library will not be checked during a normal ModelCHECK run. This is performed by writing an if-else statement containing the Windchill location of the commercial library components and the special checker group that by default run the ModelCHECK. This list can be found in a different "groups" file in the ModelCHECK directory. The group contains only the modeling system administrators. This is done as the new library components are checked thoroughly and even ModelCHECK has recently been started to be used partly from the influence of this thesis work.

6.4.2 Start file (.mcs)

Most important configurations in start file are default model names, model units, family table parameters and datum names. The start file has been set up to identify faulty case of model naming, in this case to prevent user for using the program default names starting with PRT or ASM. Also, the system of units both for mass and length are checked. Even though there

are some special parts that are modeled in imperial units, the default set is to recognize having imperial units as an error. The company has a set standard for family tables to include certain parameters and this can be verified by adding these parameters to the part file. Additionally, the datum plane names are checked to avoid system default names. PTC ModelCHECK does not directly support this but by using datum rename functionality, default named datums can be identified from the models.

6.4.3 Check file (.mcc)

Most of the ModelCHECK configuration is done in the check files. To help the configuration and trackability of the check parameters, the file can be divided in to four sections: general checks, part only checks, assembly only checks and drawing checks. The last one will be ignored in this part since this is outside of the scope of the thesis.

Starting from general verification metrics, completeness of the models is checked. Check parameters for models include checking for missing or failed components, suppressed or incomplete features, verifying family tables, geometry checks and insert mode check. ModelCHECK and Creo can identify components missing for work folder or components failing and inform user about this. This is also done automatically by the software as model is opened but where MC brings benefits is for example in large assemblies where failed components might be time consuming to find. Incomplete features can also be identified. These are simply features that were not finished accordingly or might miss references. Suppressed features are not as clear error in modeling but they are still included in the report as warnings. Suppressed features can be used correctly for example with changeable geometry in family tables. If model has a family table, all the instances can be verified. This means that the instances can be generated and do not produce critical errors. Insert mode (figure 16) can be used to quickly move backwards in the model tree. All the features and references under the Insert pointer are automatically suppressed. Insert mode left on in models will result in incomplete structures.

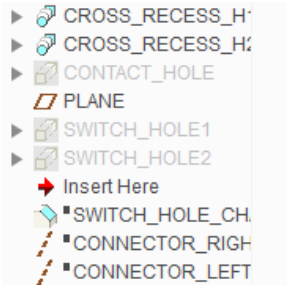


Figure 8. Insert functionality in Creo. Black box next to the icon means the that modeling operation is suppressed.

Consistency of the model in the set ModelCHECK configuration is done mostly by verifying correct start parameters, datums, family tables parameters and system of units. Model and datum name and family table parameters are checked based on the start file. The start file also defines the correct system of units. As mentioned before, different set of check can be constructed for different unit systems. However, as the imperial unit model are a very special case, only units approved by the checks are mm and kg.

A big part of the check parameters focus on the modifiability of the models. For model robustness, illegal references are checked. These include features or other references to referencing edges, chamfers and rounds. One of the most critical illegal reference is circular references. This happens when a part or assembly is referencing to itself, resulting in poor configurability among other things. Referencing to rounds, chamfers and edges is done by exploring the child relationships of the features. Also reported are external dependencies and dependent features. External dependencies happen when references are taken from external models or model need a reference from lower level in the assembly(?). Dependent feature is a feature that is created by mirroring other features. Their geometry will be based completely on the other features and these references can often cause problems when configuring the model. MC also checks for cosmetic and buried features. Cosmetic features are features that only have cosmetic effects on the model. Buried features have geometry that that is hidden under another features geometry, thus becoming redundant. Buried features result in unnecessary larger files, longer regeneration time and unexpected generation errors often due confusing references.

Flexibility of the model is verified by checking the relations configurations are correct. MC checks for relations errors, missing relations and dimensions and relations that need to be updated. Changeability is ensured by checking assembly features, bulk items and buried components. As per company's guidelines, assemblies should not contain any features in addition to parts and datums. Assembly features check can be used report any features that violates this rule. Bulk item in Creo can be used to represent non-solid bulk items such as glue that needs to be included in the Bill-of-Materials.

Some aspects of conveying design intent are also checked. This is done by checking the Creo relations functionality's rules for comments. Parameter checks that each relation rule has at least one comment line. This checks is not optimal but it can be used to remind designers to leave comments to their designs to help future re-use and modification. Other checks included in the configuration are list of flexible or frozen components, childless datum planes and generic components used in assemblies. Also, the density parameter of the model is checked and verified that it is not the default value of the modeling software. Additional information can also be reported such as total number of components. Check for verifying the start file configuration is also in effect.

One of the most difficult aspects in the check configuration file customization was to choose the correct check reporting parameters between Yes, Error and Warning. Some of the parameters have only options for Yes or No, in which case the specific check is shown as a list or a quantified number. In this stage, most of the check parameters are configured as warning as a process decision. Even though the failed model check does not lead in any actions, having 50/60 checks as warning is a safe bet.

6.4.4 Other files

The config files used in this thesis are the company's defaults and do not contain any settings that directly effects the metrics involved. They are for example used for configuring data paths and other aspect not directly effecting the actual verification. However, few options such as advanced analysis of buried features need to be turned on in configuration files if needed. One possibility in config_init file is to use an option to add a parameter for the number of errors in the models. However, this was deemed not necessary. Geometry check threshold configured in its own config-file were left untouched. These checks can provide

an overview of the geometrical quality of the models and is mostly based on VDA standard. The same was done with constant-file, where thresholds for some check such as short edges were defined. In testing throughout the thesis, these thresholds seemed adequate and can be refined further with bigger datasets if needed. Status file is used for defining the traffic light outcome of the report. It was set up to report passed as green light when there are 0 errors and maximum of two warning. The warnings are tolerated since there are few check parameters configured as warning that do not necessary cause hazards for design quality. The yellow traffic light symbol is triggered with maximum of one error and four warnings, Everything else results in red light. Furthermore, as was mentioned earlier, group-file was modified slightly by adding the commercial library admins to the list of checkers to enable running MC on components in the commercial component library.

6.4.5 Possible future customizations

Multiple other checks parameters were identified that cannot be used at this time because of lack of common methods, poor quality. One check that could be utilized in the future is checking whether the model has a raw material defined. However, this requires better and material bank and instructions for setting the correct material to be utilized in all the units. The check could potentially be already used in some settings.

One convenient functionality is parameter renaming. This checks the model for outdated parameters listed in start file and can be configured to rename the parameter according to new definitions. The same can be done for datums. This can be used for example to repair the use of outdated start files. However, these changes need to be studied further before implementing it in the production environment to make sure that the existing assemblies are not broken because of name changes.

Checking the interference between models in assemblies would also be a good feature to have. With the current model quality where older models have interfering parts that are solid this option cannot be used very effectively. However, the current guidelines already have taken interfering features into consideration and this should be implemented sometimes in the future. Checking and standardizing layers and for example simple representation naming can also be implemented in the future.

Drawing and sheet metal specific checks will also be studied further after this thesis to implement the ModelCHECK for these use cases also. This would be beneficial for example with configurable product drawings. In the case of sheet parts, the PTC RuleCHECK tool can also be used to define standard thicknesses and radiuses for example.

6.5 Case study models

Different models were selected to test the verification tools and the developed metrics with real life cases. Models selected represent assemblies and parts. The main study assembly includes subassemblies and parts that are either normal solid parts or sheet parts. The assembly model also includes so called commercial components such as bolts. These components can be skipped in ModelCHECK. The verification of these simple individual parts is also investigated as this could be easily and quickly tested and implemented in commercial component modeling verification.

First test component for the verification tool was chosen to be a low voltage switch that is a part from the commercial component library (figure 17). This was chosen because the model

is made by subcontractors in China and the model quality is then verified by commercial component library administrators. Furthermore, the improvement of the quality of so called library component would have a significant impact on the overall model quality in the company since they are used in multiple assemblies and in different platform. The model has not been released for production purposes and is currently considered a draft.

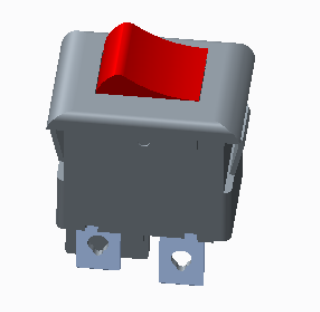


Figure 9. Commercial low voltage switch.

A small assembly was also chosen to review the effect on verification. Chosen assembly is a guiderail bracket (figure 18). This is a simple component that is used for attaching the elevator guiderails to the wall of the shaft. It mainly made of sheet metal parts and it also contains some commercial library components. This model is used in a modeling tool development platform model and thus is not in actual production use.

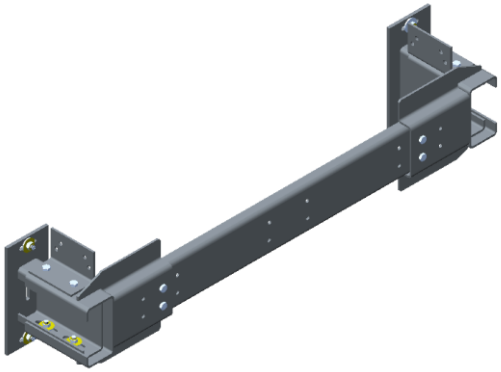


Figure 18. Guiderail bracket.

For a more complicated verification test a sling structure was chosen (figure 19). Sling is a component in elevator that houses the actual elevator car and connects to the ropes and guiderails. The specific sling was chosen because of its fairly complex structure. The sling consists of 1083 parts and sub-assemblies. These types of assemblies are practically impossible the check manually and this example showcases the most suitable use case for automated verification tool. The sling model is currently in a released lifecycle state for production use.

Figure 19. ISCS 13 Sling.

7 Conclusions

7.1 Product model verification metrics

The importance of product model quality is catching up with the product quality. In multi-system engineering environments and with engineering processes such as PLM and MBD, the product model quality is significant. Also, configurable products must be on a certain level of quality in order to bring full benefits of massconfigurable or even engineer-to-order products. Future trends in engineering are pointing towards increased use of downstream applications and sophisticated processes such as digital twins. This means that the quality demands for the models are also increasing.

First of the research questions in this thesis was what metrics can be used to evaluate and verify product model? Literature research presented multiple different sets of product model quality dimensions and metrics. Similarities between authors can be found and most used key aspects were model accuracy. consistency, completeness, timeliness and accessibility. These can be used as basic evaluation metrics for model quality when models are fixed. However, with configurable models more metrics need to be defined for modifiability, re-usability, configurability and interoperability. Literature also provides answers and deeper metrics for these aspects of models. For example, management of references and model tree are significantly beneficial for model modifiability and conveying design intent. Overall quality is not easily quantifiable as is. Manual human inspection will still be a necessity for years to come, at least in the case of certain quality aspects and with complex model.

Based on literature research and interview inside the company, metrics for evaluating product model quality were developed. The metrics were divided to general verification metrics and product modifiability metrics. The metrics found in this thesis are still subject to change and fine tuning will be needed to yield the best possible benefits.

Defining and measuring the quality in product models is always an ongoing process. Model quality improvement should be seen as an iterative process. The process should start with defining the quality aspects and then continue into measuring them. Measurement data also need to be analyzed and fixing procedures should be implemented. Studies in the field suggest that more theoretical basis in need in the future for more complex quality requirements. Some metrics for shape errors have been identified but further improvements are still needed. Furthermore, for best results companies and instutions need to research their own needs for product model quality and develop and focus on implementation of their own metrics.

7.1.1 Verification automation

The second research question was can product model verification be automated? Multiple methods and model quality tool were found during the literature research. Some of the model quality tools are embedded into the modeling software and some of them are external add-on software. Some notable verification methods involve history-based, rule-based and formal methods.

Benchmarks of companies using model quality tools for automated product model verification was also found. The companies report benefits of overall product model quality with the use of model checking tools. One important aspect with product quality tools is the cost. Embedded model checkers have significantly smaller costs and might even be included with

a corresponding modeling software license. Utilizing tools already in the modeling environment is also a considerable saving for both time and money.

PTC ModelCHECK was chosen for the product model verification automation tool in this thesis. This was based on the availability of the software and a possibility of rapid implantation for production use. The checking parameters were based on the literature research, interviews and the developed metrics. Some of the checks include geometry checks, datum naming, bad referencing, bad child-parent relationships and model rule comments. Verification rules and measurement always require customization to be optimized. Quantitative metrics exist for identifying product model shape defects but these metrics are somewhat context dependent and measured subjectively in each different model quality tool.

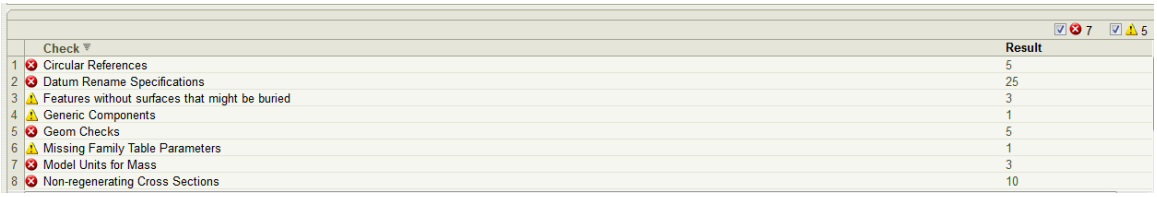
The ModelCHECK configuration created can catch errors in different quality aspects of the model but still does not offer definite answers if the model quality is high enough for production or downstream applications. However, limiting and eliminating errors automatically without a significant loss of time is very valuable. Furthermore, ModelCHECK can still be used as a reliable indicator of models quality and combined with manual inspection of some aspects that cannot currently be automated, the quality of models will improve and the time used for verification will decrease. Product model verification tools such as ModelCHECK are still quite restricted for example in the terms of conveying design intent. For development of more sophisticated verification tool, better quantitative metrics and more advanced quality tool development platform is needed.

7.1.2 Case study results

Common company issued 3D modeling work laptop was used for the ModelCHECK tests. The laptop has a 2-core/4-thread processor, dedicated Quadro graphics card and 8 gigabytes of RAM with Windows 7 Enterprise operating system. The modeling software used was Creo version 2.0 M200 with some company add-ons installed.

ModelCHECK reports three errors/warnings with the low voltage switch. MC identifies buried features, unnamed datums and missing family table parameters in the model. Identifying the datum and parameters errors in this case is quite easy for the person verifying the commercial component models. However, the buried feature is hidden under other geometry and thus hard to notice without manually checking all the features. The small bracket assembly does not produce any errors in the check. This might be a result of simple components used in the assembly.

In the sling assembly, ModelCHECK detects multiple defects and modeling errors (figure 20). Most significant errors are circular references, wrong system of units, external dependencies and generic components in assembly. Others errors include failed geometry checks, unnamed datums and relation errors and warnings. These errors will most probably cause problems in the lice-cycle of the product model if they are not fixed. Even though the model in question might produce desired configuration results, it cannot be considered as high quality or re-usable with the about of quality defects.



	Check	Result
1	Circular References	5
2	Datum Rename Specifications	25
3	Features without surfaces that might be buried	3
4	Generic Components	1
5	Geom Checks	5
6	Missing Family Table Parameters	1
7	Model Units for Mass	3
8	Non-regenerating Cross Sections	10

Figure 10. Some of the quality defects found in the sling model.

As can be seen from the test cases, the benefits are most significant in small parts with fairly complex geometry or in large complex assemblies. The time saving when checking the certain qualities in the large assembly is notable significant. Even running the configured check on the large assembly took only around one minute and 45 seconds. With the smaller assembly and the commercial component, the test only took few seconds. In design time frame, the time used for running ModelCHECK is negligible and arguably necessary for possibility to achieve high quality product data in CAD environment.

Overall, the goals of the thesis for automated product model verification were achieved. ModelCHECK configuration presented in this thesis can be used for a part of the evaluation of a product model. Designer verification is still needed for all-encompassing quality check of the product as model quality tools are mostly limited to recognizing low-level semantic issues in the models.

7.2 Recommendations for future work inside company

For ModelCHECK development, few future steps were identified. The refinement of check parameter outputs between warning and errors need to be done after a larger sample sizes. To fully help designers utilize the tools presented, help documentation and use instruction need to be on a good level. Furthermore, expanding ModelCHECK for 2D drawing verification should also be pursued and there have already been some initial talks about this. With increased model quality, the possibility of using stricter and additional checks should also be investigated as the process ages. One recommendation is to implement ModelCHECK also in the CWS tool. This way, the effects on configurability of the models could potentially be examined. Also, utilizing PDM verification tool in CWS would be recommended.

Both the development of metrics and verification automation should be continued as an iterative process as the methodology and modeling conventions evolve. The metrics and verification tool developed in this thesis should provide a solid baseline for future development. On current PDM environment, more effective use of existing verification tools such as release validator should be examined. Implemented stricter checks one-by-one might be a possible solution for minimizing the resistance coming from the design field. During the research, multiple different verification tools for different purposes were identified. Verification of BIMs and exchange verification between formats can also be based on some of the work done in this thesis.

A big part of improving quality of the product models is to raise awareness about the issues in modeling and how these issues can be solved or avoided. Creating notifications and training for common errors could decrease the number of these errors being made while modeling. The quality aspects presented can also be used for updating the modeling and verification guidelines.

8 Summary

Product model quality is lifting its position as one of the most important aspects in modern engineering system consisting of multiple design environments and downstream applications. The main goals for this thesis are finding verification metrics for evaluating product model quality and research and development of automated product model verification.

Parametric and feature-based CAD program are nowadays often used in engineering systems along with PDM and PLM solutions. Also, the use of simulation and calculation software based on product model data has increased in recent years. With systems consisting of multiple design environments and file formats, the importance of common data throughout the system is undeniable. Some neutral formats can be used to relay the data between systems but no real all-encompassing standard exist for the purpose. The communication between systems can also happen on software level, for example via APIs. Product models are nowadays often configurable which enables effective design of diverse product range. However, configurability also requires additional elements compared to traditional non-configurable product models. In the future, even more complex product model structures are expected and the emergence of using product models for downstream applications will increase the quality standards.

Errors in product models often stem from the user, modeling software, databases and modeling methodology. Quality can be influenced by modeling methodology, verification and repairing of the models. As mentioned the importance of model quality has been steadily increased in the recent years. This can be seen in the research done concerning quality aspect of models. Verification and validation are often used interchangeably despite their different paradigms. Verification can be defined as inspecting if the model if created in a correct way and validation as inspecting if the correct model was created. Different verification methods such as satisfactory problem based and rule based were identified from the literature research. Studies have also been done for verification tools used for data and CAD data. The tools can be used for automation of product model or product data verification. Furthermore, benchmark for the use of verification tools and automated verification also were researched.

Multiple product model quality dimensions can be identified from literature. Some of the most common aspects are accuracy, timeliness, consistency, completeness and availability. Configurable and modifiable product models need additional quality dimension considered when measuring their quality. Good modeling implementation on dimensions such as simplicity, robustness, flexibility of structures and conveying design intent create better modifiability. These aspects will increase the model capability of configuring, re-using and interoperability. These dimensions can be broken down further for metrics such as quality of references and model tree structures.

Current verification processes inside the company were researched by interviewing different sectors in the company from Finland and Italy. Common needs for verification work were identified to be better evaluation of configurability, better re-use of models and better data for downstream applications. Furthermore, the verification tools available in the current modeling environment were studied as potential development platforms for verification automation tools.

The platform for development of automated verification tool was chosen as PTC ModelCHECK. It was chosen because it is readily available in the modeling system but is

not fully utilized. Furthermore, the tool should be in line with the metrics developed in this thesis. Setting up customized ModelCHECK was done by editing the configuration files by adding and removing check and start parameters. Some of the most important check were identified to be catching bad references, problematic child-parent relations, correct start parameters and units and buried features. The complete list of check parameters and their explanations can be found in the appendix.

Three test components were chosen for testing the ModelCHECK configuration. From the test components, the bracket assembly passed the checks without any error identified. However, the two other models did not. Firstly, a low voltage switch model triggered three errors or warning that affect the product quality negatively. Secondly, the biggest test component, a sling assembly had several critical errors founds. These errors include wrong system of unit, regeneration errors, relation errors and circular references. The check itself can be considered time saving as even the biggest assembly was checked within two minutes.

Overall, the goals of the thesis were met. A set of dimensions and metrics were presented for product model quality. These can be used for evaluating the quality of models and as a starting point for further development. The automated verification tool was developed for the embedded verification tool used in the design environment. The tool was discovered to be effective and time-saving in finding certain errors or bad modeling methods in CAD models.

References

[1] Laakko, T. Tuotteen 3D-CAD-suunnittelu. Helsinki ; Porvoo ; Juva : WSOY, 1998. 311 p. ISBN 951-0-23217-3

[2] Schoonmaker, S. J. (2002). The CAD guidebook: A basic manual for understanding and improving computer-aided design (Vol. 150). CRC Press.

[3] ISO 10303-108. Industrial automation systems and integration—Product data representation and exchange: Integrated application resource: Parameterization and constraints for explicit geometric product models. Geneva (Switzerland): International Organization for Standardization; 2005.

[4] Shah, J. J., & Mäntylä, M. (1995). Parametric and feature-based CAD/CAM: concepts, techniques, and applications. John Wiley & Sons.

[5] Yang J, Han S, Kang H, Kim J. Product data quality assurance for e-manufacturing in the automotive industry. Int. J. Comput. Integr. Manuf. 2006; 19(2): 136–47.

[6] Stark, J. Product lifecycle management. In Product Lifecycle Management Vol.1 pp. 1-29. Springer International Publishing, 2015.

[7] Peltonen, H. PDM: tuotetiedon hallinta. Helsinki, Edita, IT Press, 2002. 169 s. ISBN 951-826-664-6.

[8] Valo, J. Varastohyllystön automatisoitu suunnittelu. Master's thesis. Espoo, Aaltoyliopisto, 2013. 68p.

[9] Pipino, L. L., Lee, Y. W., & Wang, R. Y. (2002). Data quality assessment. Communications of the ACM, 45(4), 211-218.

[10] Luteberget, B., Johansen, C., & Steffen, M. (2016, June). Rule-based consistency checking of railway infrastructure designs. In International Conference on Integrated Formal Methods (pp. 491-507). Springer, Cham.

[11] Zhang, J., & El-Gohary, N. M. (2017). Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking. Automation in Construction, 73, 45-57.

[12] Autodesk. Product configurator. Available from URL: https://damassets.autodesk.net/content/dam/autodesk/www/products/autodesk-configurator-360/fy18/overview/images/product-configuration-large-1152x865.jpg. [retrieved 09-11-2017].

[13] Osorio, J., Romero, D., & Molina, A. (2013). A Modeling Approach towards an Extended Product Data Model for Sustainable Mass-Customized Products. IFAC Proceedings Volumes, 46(9), 579-583.

[14] Abdel-Malek, K., Zou, H. L., Wang, J. Y., & Othman, S. (1999). Automated design and parametrization of mechanical part geometry. Research in engineering design, 11(4), 206-217. ISSN 1435-6066.

[15] Gosling, J., & Naim, M. M. (2009). Engineer-to-order supply chain management: A literature review and research agenda. International Journal of Production Economics, 122(2), 741-754.

[16] Hirz, M., Rossbacher, P., & Gulanová, J. (2017). Future trends in CAD–from the perspective of automotive industry. Computer-Aided Design and Applications, 1-8.

[17] Slideshare. Using Hadoop to build a Data Quality Service for both real-time and batch data. Available from URL : https://www.slideshare.net/HadoopSummit/using-hadoop-to-build-a-data-quality-service-for-both-realtime-and-batch-data [retrieved 15-01-2018].

[18] Wand, Y., & Wang, R. Y. (1996). Anchoring data quality dimensions in ontological foundations. Communications of the ACM, 39(11), 86-95.

[19] AIAG, D. (2001). ISO/PAS 26183:2006, SASIG Product data quality guidelines for the global automotive industry. Available from URL:http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43436; 2005 [retrieved 09-11-2017].

[20] Boyd, M. (2009). Product Data Quality - Different Problem, Different Solutions. MIT Information Quality Industry Symposium, July 15-17, 2009. Available from URL: http://mitiq.mit.edu/IQIS/Documents/CDOIQS_200977/Papers/02_08_2D-2.pdf [retrieved 09-11-2017].

[21] Finn, G.(2000). Building quality into design engineering. Quality digest. Available from URL: https://www.qualitydigest.com/feb00/html/design.html [retrieved 09-11-2017].

[22] Pecheur, C., & Nelson, S. (2002). V&V of Advanced Systems at NASA. NASA ARC. Technical Report NASA/CR-2002-211402.

[23] Eckerson, W. W. (2002). Data quality and the bottom line: Achieving business success through a commitment to high quality data. The Data Warehousing Institute, 1-36.

[24] Redman, T. C. (1998). The impact of poor data quality on the typical enterprise. Communications of the ACM, 41(2), 79-82.

[25] ERIM, Center for Electronic Commerce, CAD/CAM Data Problems and Costs in the Tool and Die Industry. Available from URL: http://www.erim.org/cec/paperscadcam_exec.htm [retrieved 09-11-2017].

[26] Trippner, D., & Endres, M. (1998). STEP–The Significance for the Designer. Product Data Journal, 2, 13-15.

[27] Moody, D. L., & Shanks, G. G. (2003). Improving the quality of data models: empirical validation of a quality management framework. Information systems, 28(6), 619-650.

[28] Contero M, Company P, Vila C, Aleixos N. Product Data Quality and Collaborative Engineering. IEEE Comput. Graphics Appl. 2002; 22: 32-42.

[29] González-Lluch, C., Company, P., Contero, M., Camba, J. D., & Plumed, R. (2017). A survey on 3D CAD model quality assurance and testing tools. Computer-Aided Design, 83, 64-79

[30] D.P. Ballou, H.L. Pazer, “Modeling data and process quality in multi-input, multi-output information systems”, Management Science Vol 31, No. 2 (1985), pp. 150-162

[31] Company, P., Contero, M., Otey, J., & Plumed, R. (2015). Approach for developing coordinated rubrics to convey quality criteria in MCAD training. Computer-Aided Design, 63, 101-117

[32] Company P. Contero M., Otey J., Camba J.D., Agost M.J. and Pérez-López D.C. (2016). Webbased system for adaptable rubrics: case study on CAD assessment. Journal of Educational Technology & Society.

[33] Wang, R. Y. (1998). A product perspective on total data quality management. Communications of the ACM, 41(2), 58-65.

[34] Yang, J., Han, S., & Park, S. (2005). A method for verification of computer-aided design model errors. Journal of Engineering Design, 16(3), 337-352.

[35] SAS. Three Critical Steps to Improving Product Data Quality. Technical white paper. Available from URL: https://www.sas.com/content/dam/SAS/en_us/doc/whitepaper1/three-critical-steps-improving-product-data-quality-106029.pdf [retrieved 09-11-2017].

[36] Son, S., Na, S., & Kim, K. (2011). Product data quality validation system for product development processes in high-tech industry. International Journal of Production Research, 49(12), 3751-3766.

[37] AIAA. (1998). Guide for the Verification and Validation of Computational Fluid Dynamics Simulations. American Institute of Aeronautics and Astronautics, AIAA-G-077-1998.

[38] Thacker, B. H., Doebling, S. W., Hemez, F. M., Anderson, M. C., Pepin, J. E., & Rodriguez, E. A. (2004). Concepts of model verification and validation (No. LA-14167). Los Alamos National Lab., Los Alamos, NM (US).

[39] Autodesk. Virtual Realities – Product Verification and Validation. Available from RL: https://www.autodesk.com/industry/manufacturing/resources/mechanical-engineer/product-verification-and-validation [retrieved 09-11-2017].

[40] Maropoulos, P. G., & Ceglarek, D. (2010). Design verification and validation in product lifecycle. CIRP Annals-Manufacturing Technology, 59(2), 740-759.

[41] McKenney, D. (1998). Model quality: the key to CAD/CAM/CAE interoperability. International TechneGroup Incorporated, Milford, OH.

[42] Denman, W. (2017). Automated verification of continuous and hybrid dynamical systems (No. UCAM-CL-TR-910). University of Cambridge, Computer Laboratory.

[43] Kukimoto, Y. (1996). Introduction to Formal Verification. Available from URL: https://embedded.eecs.berkeley.edu/research/vis/doc/VisUser/vis_user/node4.html [retrieved 09-11-2017].

[44] Jeongsam Y, Han S. Repairing CAD model errors based on the design history. Computer-Aided Design. (2006). Vol. 38:6. S. 627-640. ISSN 0010-4485. Saatavissa DOI 10.1016/j.cad.2006.02.007.

[45] Dantan, J. Y., & Qureshi, A. J. (2009). Worst-case and statistical tolerance analysis based on quantified constraint satisfaction problems and Monte Carlo simulation. Computer-Aided Design, 41(1), 1-12.

[46] Schamai, W., Albarello, N., Helle, P., Buffoni, L., & Fritzson, P. (2017). Towards the Automation of Model-Based Design Verification. INSIGHT, 20(1), 42-48.

[47] Eastman, C., Lee, J. M., Jeong, Y. S., & Lee, J. K. (2009). Automatic rule-based checking of building designs. Automation in construction, 18(8), 1011-1033.

[48] Perera, N. (2016). Automatic Configuration Management: Autodiscovery of Configuration Items and Automatic Configuration Verification. In 14th International Conference on Space Operations.

[49] Gerace, J. J. (2013). Understanding verification and validation of product model data in industry (Doctoral dissertation, Purdue University).

[50] Barateiro, J., & Galhardas, H. (2005). A survey of data quality tools. Datenbank-Spektrum, 14(15-21), 48.

[51] NASA. (2012).Take the Reins on Model Quality with ModelCHECK and Gatekeeper. Available from URL: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20120004042.pdf [retrieved 09-11-2017].

[52] Vesecký, J. More efficient product development process through the quality of CAD data. PTC LifeWorx Europe 2015, Stuttgart,17-18 November 2015.

[53] ITI Global. CADIQ enables Six Sigma model quality. Available from URL: https://www.iti-global.com/uploadIMG/moxie/Case%20Studies/Ford-Case.pdf [retrieved 09-11-2017].

[54] Rautio, T. 2017. Component CDE, Hoisting mechanics. KONE. Hyvinkää, Myllykatu 3. Interview 5.4.2017.

[55] Mäkelä, K. Solution Design Owner, MCAD. KONE. Hyvinkää, Myllykatu 3. Interview 6.4.2017.

[56] Vestman, V. Senior Engineer, Mechanics. KONE. Hyvinkää, Myllykatu 3. Interview 6.4.2017.

[57] Gallucio, F. Car Project Manager. KONE. Hyvinkää, Myllykatu 3. Interview 6.6.2017.

[58] Rodriguez-Toro, C. A., Tate, S. J., Jared, G. E. M., & Swift, K. G. (2003). Complexity metrics for design (simplicity+ simplicity= complexity). Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 217(5), 721-725.

[59] Zäh, M. F., Möller, N., & Vogl, W. (2005, September). Symbiosis of changeable and virtual production–the emperor's new clothes or key factor for future success. In Proceedings (CD) of the international conference on changeable, agile, reconfigurable and virtual production, Munich, Germany.

[60] Li, M., Zhang, Y. F., Fuh, J. Y., & Qiu, Z. M. (2011). Design reusability assessment for effective CAD model retrieval and reuse. International Journal of Computer Applications in Technology, 40(1-2), 3-12.

[61] Mun, D., Han, S., Kim, J., & Oh, Y. (2003). A set of standard modeling commands for the history-based parametric approach. Computer-aided design, 35(13), 1171-1179.

[62] Krogstie, J. (2015). Capturing enterprise data integration challenges using a semiotic data quality framework. Business & information systems engineering, 57(1), 27-36.

[63] Aleixos, N., Company, P., & Contero, M. (2004). Integrated modeling with top-down approach in subsidiary industries. Computers in Industry, 53(1), 97-116.

[64] Batini, C., Cappiello, C., Francalanci, C., & Maurino, A. (2009). Methodologies for data quality assessment and improvement. ACM computing surveys (CSUR), 41(3), 16.

[65] Jinsong, Z., Qifu, W., Li, W., & Yifang, Z. (2005). Configuration-oriented product modelling and knowledge management for made-to-order manufacturing enterprises. The International Journal of Advanced Manufacturing Technology, 25(1), 41-52.

[66] PTC. ModelCHECK - Help Topic Collection - PTC Community. Available from URL: https://community.ptc.com/sejnu66972/attachments/sejnu66972/CreoBlog/687/1/modelcheck.pdf [retrieved 09-11-2017].

Appendices

Appendix 1. Questionnaire for product model quality. 1 page.
Appendix 2. Custom ModelCHECK files and explanations. 6 pages.

Appendix 1. Interview questions about product model quality and verification

This questionnaire focuses on verification that can be defined as "The process of determining that a model implementation accurately represents the developer's conceptual description of the model and the solution to the model". In other words, it answers the question "Is the model created in a right way?"

Product model = 3DCAD + PDM model

Model Quality

How is the model quality managed in your products? (Whose responsibility, who does the verifying?)

Are you satisfied with the current product model quality?

Do you have guidelines for model quality/verification? (in addition to KOS-000065 and 51061748D01)

How do you feel the currently used modelling guidelines effect the model quality?

Do you log the occurring errors/defects in models? If so, how?

Do you think there would be benefits in implementing an automated verification procedures or tool?

1) Model verification

How do you verify 3DCAD models? (Geometry, structure, rules, modularity, attributes)

How do you verify PDM models? (Structure, rules, configuration, attributes)

Do you verify something else from product models?

Do you use any tools for model verification? (PTC ModelCHECK, VariPDM functionalities, etc.)

Is there need for more functionalities in the used tools? If so, what?

Are there any checks you would like to do but can't at the moment?

Verification metrics

What are the must-do checks when verifying models?

How do you evaluate the model quality on changeability (configurability, re-usability, robustness and modularity)?

How are the test cases selected for configuration validation?

Does your product/unit have special metrics that are used in evaluating the model quality?

Are there additional aspects that need evaluation in the product models?

Appendix 2. Custom ModelCHECK files and explanations

The parameter descriptions in this appendix are based on PTC ModelCHECK help topic[66].

Conditions file:

```
# SET CONFIG FILE
IF ( MODEL_UNIT EQ MM ) config=(check/Kone_simple.mch)(start/Kone_defaults.mcs)(constant/mm.mcn)(status/sample_status.mcq)
        ELSE config=(check/Kone_simple.mch)(start/Kone_defaults.mcs)(constant/mm.mcn)(status/sample_status.mcq)
```

Checks model unit is equal to mm. If unit yes, ModelCHECK runs with parameters set in these config files. Note that if unit is not mm, model check runs with the same default files.

```
# OVERRIDE CHECKS
IF ( PTC_WM_CONTEXT_ID EQ OR:wt.inf.library.WTLibrary:27410 ) AND ( GROUPNAME NEQ CHECKER ) NOCHECK
IF ( PTC_WM_LIFECYCLE_STATE EQ Retired ) NOCHECK
```

Checks if PTC_WN_CONTEXT_ID parameter is equal to specified value (OR:wt.inf.library.WTLibrary:27410). If both are true, ModelCHECK is not run on the model. The second condition checks the life-cycle state of the models and skips the check with model that have been retired.

Start file:

```
# PART MODE START PART REPORT CONFIGURATION
PRT_MODEL_NAME                  NEQ PRT*
PRT_UNITS_LENGTH    MM
PRT_UNITS_MASS      KILOGRAM
PRT_FT_PARAMETER                DESCRIPTION
PRT_FT_PARAMETER                OBJECT_ID
PRT_FT_PARAMETER                DOCUMENT_ID
PRT_FT_PARAMETER                PRO_MP_SOURCE
PRT_FT_PARAMETER                PRO_MP_ALT_MASS
```

Defines that part name should not equal to PRT*(for example PRT001 can't be saved). Defines the default unit length to mm and mass to kg. These paramaters define the required family table paramaters for models.

```
#Check for unnamed datums
PRT_DATUM_RENAME DTM1 UNNAMED_DATUM
PRT_DATUM_RENAME DTM2 UNNAMED_DATUM
PRT_DATUM_RENAME DTM3 UNNAMED_DATUM
PRT_DATUM_RENAME DTM4 UNNAMED_DATUM
PRT_DATUM_RENAME DTM5 UNNAMED_DATUM
PRT_DATUM_RENAME UNNAMED_DATUM DTM1
```

Checks for datums in the model named DTM1-5 and suggest a new name of unnamed_datum. Last line check if there datums named unnamed_datum and suggest a new name of DTM1 by default. This eliminates a possibility of accidently leaving a "unnamed_datum" in the model after initial run with ModelCHECK.

```
# ASSEMBLY MODE START PART REPORT CONFIGURATION
ASM_MODEL_NAME                  NEQ ASM*
ASM_UNITS_LENGTH    MM
ASM_UNITS_MASS      KILOGRAM
ASM_FT_PARAMETER                DESCRIPTION
ASM_FT_PARAMETER                OBJECT_ID
ASM_FT_PARAMETER                DOCUMENT_ID
ASM_FT_PARAMETER                PRO_MP_SOURCE
ASM_FT_PARAMETER                PRO_MP_ALT_MASS
```

Defines that assembly name should not equal to ASM* (for example ASM001 can't be saved). Defines the default unit length to mm and mass to kg.

```
ASM_DATUM_RENAME ADTM1 UNNAMED_DATUM
ASM_DATUM_RENAME ADTM2 UNNAMED_DATUM
ASM_DATUM_RENAME ADTM3 UNNAMED_DATUM
ASM_DATUM_RENAME ADTM4 UNNAMED_DATUM
ASM_DATUM_RENAME ADTM5 UNNAMED_DATUM
ASM_DATUM_RENAME UNNAMED_DATUM ADTM1
```

Does the same check as with prt_datum_rename but with assembly model and with default assembly datum names.

```
# DRAWING INFORMATION
DRW_MODEL_NAME                  NEQ DRW*
MC_REGEN_CONFIG_FILE   text/mc_regen.mcr
MCSI5_TOL          .03
```

Defines that drawing name should not equal to .drw (in other words, the name is not empty). Defines the regeneration config file for drawings.

Check-file parameters for regeneration check:

AF_INCOMPLETE *W*

Incomplete annotation features. Reports whether the annotation feature in the part is incomplete. The annotation feature is incomplete when strong references of an annotation element are missing.

ANNTN_INACTIVE *W*

Checks for inactive annotations within a model and allows you to redefine the annotation feature that has the annotations. ModelCHECK also allows you to delete the inactive annotation element.

EDGE_REFERENCES *E*

Lists features that have been created using edges as dimension reference points. If any are found, you can highlight them in the Pro/ENGINEER window.

FAMILY_INFO *E*

Identifies a model as either generic or not. If the model is generic, this check reports the instances with their names, and whether the instances have been successfully verified.

FILE_SIZE *Y*

Displays the disk space that is used to store the model.

FT_DEF_VALS *W*

Family table default values. If the model has a family table, this check makes sure that no instances have default values [there are no asterisks (*) in the table].

IGNORE_FEAT *Y*

Problem features that should be ignored. Sometimes it is necessary to create a feature that ModelCHECK will view as an error. ModelCHECK can be configured to ignore the problems. Problems can be set to the ignore state from any place that allows highlighting of problems.When ModelCHECK runs on a model that contains ignored features, the ignored features are listed in the ModelCHECK report and can be highlighted. From the report, the status of the ignored features can be reset so that ModelCHECK resumes warnings about the problem.

INCOMPLETE_FEAT *E*

Reports whether any incomplete features exist in the model. If any are found, you can highlight or delete them in the Pro/ENGINEER window. ModelCHECK regenerates the model if any incomplete features are deleted.

INSERT_MODE *E*

Reports whether the Insert mode is active.

MODEL_NAME *W*

Verifies that the names of parts and assemblies conform to the standard naming conventions defined by the PRT_MODEL_NAME and ASM_MODEL_NAME start configuration options.

REGEN_ERRS *E*

Reports any errors when a model is fully regenerated.

REGEN_WRNS *W*

Reports any warnings when a model is fully regenerated.

RELATION_ERRS *E*

Checks for errors in the model's relations and reports the relation lines containing errors.

RELATION_MISS *W*

Checks for standard relations and their comments in parts and assemblies. If any are missing, ModelCHECK adds them to the model. Standard relations and comments are listed in the configuration files using the PRT_COMMENT, PRT_RELATION, ASM_COMMENT and ASM_RELATION start configuration options.

RELATION_UPDATE *W*

Checks for relations in the model that need to be updated. Relations are defined using the RELATION_UPDATE_FILE configuration option in the start configuration file.

RELATION_COMM *W*

Checks that every relation has at least one comment line.

RELATION_MULT *W*

Checks that no dimensions and parameters have been assigned multiple times in the relations file.

SUP_FEATURES *W*

Lists the types and IDs of the suppressed features in the model. Any features that are included in family tables or sheet metal flat-pattern features are ignored.

UNITS_LENGTH *W*

Checks that the length units are from a standard list of acceptable units. Standard length unit types are designated in the start configuration file.

UNITS_MASS *W*

Checks that the mass units are from a standard list of acceptable units. Standard mass unit types are designated in the start configuration file.

EXTERNAL_DEPS *W*

Lists all the external dependencies of a model. The names of the assemblies in which the external dependencies exist are also listed. You can highlight the external dependencies.

PLANE_CHILD *W*

Reports any datum planes in the model, other than default planes, with no children. In the start part list in the start configuration file, you can specify a list of required standard datum planes for a model. The PLANE_CHILD check does not check these standard datum planes.

FT_STD_PARMS *W*

Ensures that the standard parameters have been added to models with family tables. Standard parameters are set in the start configuration file.

MCREGEN_VERIFY_FT_INSTS *Y*

Verifies all instances during ModelCHECK Regenerate if the assembly is a generic representative of a family.

DEPENDENT_FEATURE *W*

Checks for the existence of dependent features. A dependent feature is created when you copy a feature by translating, rotating, or mirroring it. The copied as well as the original features are reported and can be highlighted.

COSMETIC_FEAT *Y*

Lists the cosmetic features in the model.

DTM_AXES_INFO *Y*

Lists all datum axes found in the model.

DATUM_RENAME *E*

If this check is enabled and a specified datum is found in the model, ModelCHECK renames it as specified. Datum names to be renamed are specified using the PRT_DATUM_RENAME and ASM_DATUM_RENAME start configuration options.

 # PART REPORT CONFIGURATION

BURIED_FEAT *W*

Reports any buried features in the model. If any are found, you can highlight them in the Pro/ENGINEER window. Buried features are completely enveloped by another feature.

FAILED_FEATURES *W*

Lists the featues that are failed.

CHILD_FAILED_FEATURES *W*

List the features that are children of failed features.

CHAMFER_CHILD *E*

Reports any features that are children of chamfers. If any are found, you can highlight them in the Pro/ENGINEER window.

DEF_DENSITY *W*

Checks that the model's density is not the default (1.00). If it is the default, update the density from the ModelCHECK report.

DRAFT_CHILD *W*

Lists the features that are children of draft features. If any are found, you can highlight them in the Pro/ENGINEER window.

IMPORT_FEAT *W*

ROUND_CHILD *E*

Lists the features that are children of rounds. If any are found, they can be highlighted in the Pro/ENGINEER window.

BURIED_SUSPECT *W*

Reports the child features of buried features in the model. Creo ModelCHECK reports BURIED_FEAT when the buried features do not have child features.

GEOM_CHECKS *W*

Reports if there are any geometry checks in the model. If any are found, the features in question can be highlighted in the Pro/ENGINEER window.

DTM_AXES_INFO *Y*

Lists all datum axes found in the model.

 # ASSEMBLY REPORT CONFIGURATION

ASM_FEATURES Y

Reports any assembly features, other than datums, in the model.

BULK_ITEMS W

Reports any bulk items found in the assembly.

CIRCULAR_REFS E

Checks for any circular references in the assembly. Circular references occur when an assembly contains a number of cross references that form a loop.

CHILD_FAILED_COMPONENTS E

Reports components that are children of failed components. Children of failed components are components with one or more parents that are failed components.

CHILD_EXT_FAILED_FEATURES E

Reports features that are children of external failed features. Children of external failed features are features with one or more external parents that are failed features or children of failed features.

CHILD_EXT_FAILED_COMPONENTS E

Reports components that are children of external failed components. Children of external failed components are components with one or more external parents that are failed components or children of failed components.

FRZ_COMPONENTS E

Checks for frozen components in the assembly.

GEN_COMPONENTS W

Reports the components in an assembly in which the generic of a family table, rather than an instance, was assembled.

MIS_COMPONENTS E

Reports missing components. These occur when Pro/ENGINEER does not know where a component, needed for an assembly, can be found on the disk.

NUM_COMPONENTS Y

Reports the total number of components in an assembly.

SUP_COMPONENTS Y

Checks for any suppressed components in the assembly.

PACK_COMPONENTS Y

Reports any components that are not fully constrained or that are packaged.

UNQ_COMPONENTS Y

Reports the number of unique components in an assembly.